

BIMESTRAL

Ano 2 • n.º 9 • Novembro 1985 • Preço 250\$00

SOFTWARE

Revista para Profissionais e Amadores de Informática

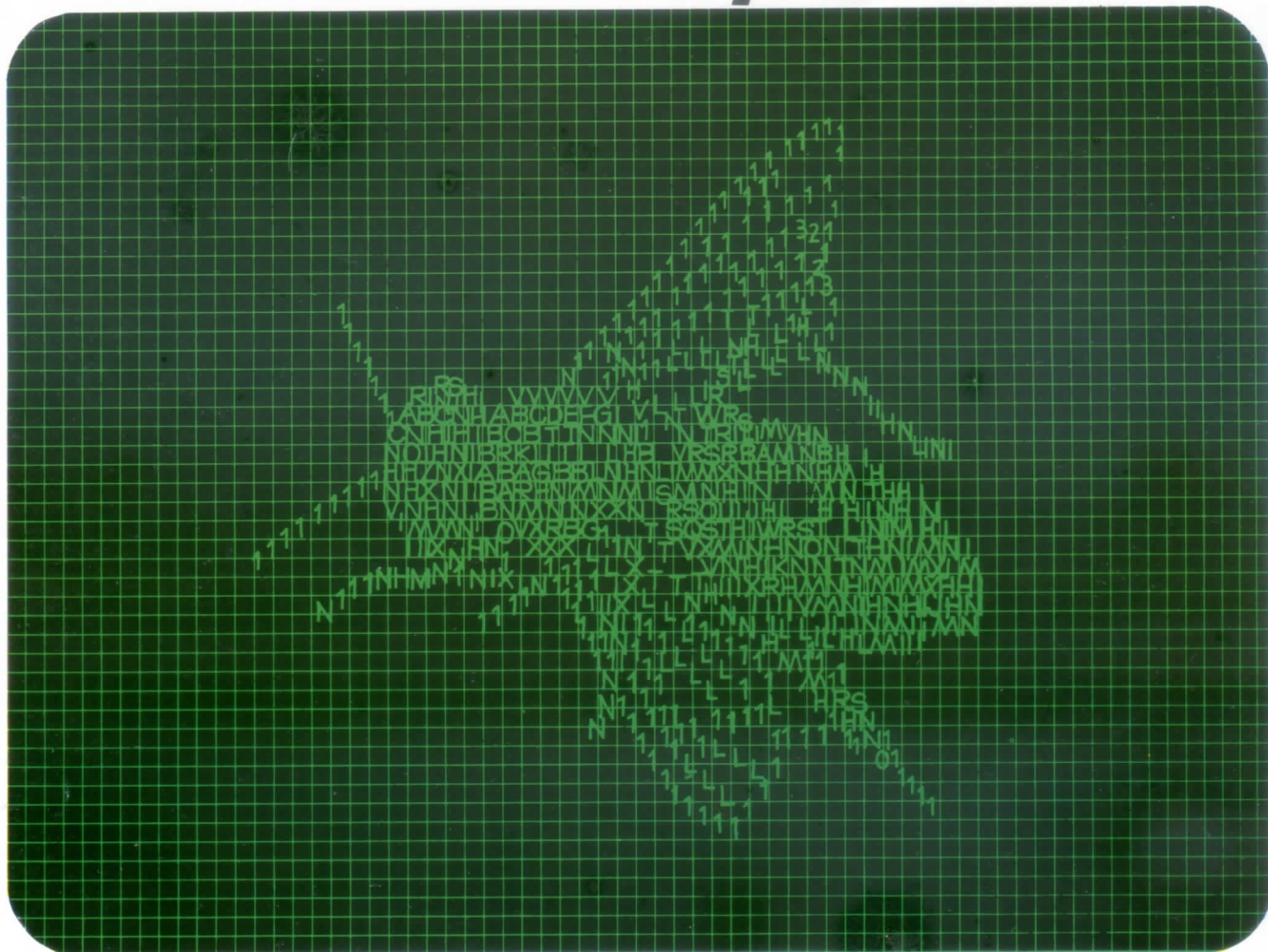
A REVISTA DE INFORMÁTICA MAIS VENDIDA EM PORTUGAL



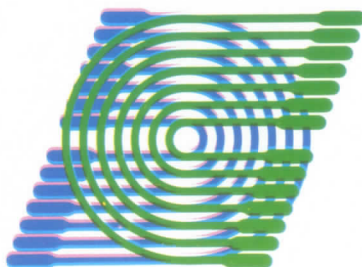
Belgica 150fb Holanda 10fl Brasil 5,000ct\$ Alemf Oc. 6dmb

Telemática

*sabe como é importante
racionalizar a produção.*



As abelhas não precisam de computador para produzir, armazenar e distribuir o produto do seu trabalho. Mas, como nos falta esse instinto profundo e maravilhoso com que a Natureza dotou as abelhas, recorremos a poderosos meios de Informática para racionalizar a produção.



A TELEMÁTICA ajuda as Empresas modernas com cálculos prévios e exactos sobre as suas necessidades, de modo a determinar os seus processos de trabalho, com menos custos e melhores resultados. A TELEMÁTICA dispõe de recursos tecnológicos e humanos que permitem a uma Empresa, tal como as abelhas, tirar o melhor partido do seu esforço.

Telemática
a solução informática



software



N.º 9 — Novembro/Dezembro

Director/Editor
Paulo Pinto

Director Executivo
Raul Mendes Pinto

Redacção
Adriano Oliveira
Vasco Ega

Secretária da Redacção
Marina Santos

Assessores Técnicos
Dr.ª Paula Ponce
António Dionísio
Carlos Amaral
Fernando Garcia

Assessor Jurídico
Dr.ª Ana Osório Nunes

**Capa
e Direcção de Design**
Arq. Luis Costa Pereira

Colaboram neste número
P. Poutois, F. Rion, G. Goeleven, K. Aziz,
Carlos Amaral, Paulo Amaral, J. Auquier,
Fernando Garcia, António Tavares, Y.
Noël-Roch, E. Dormont

**Redacção, Administração
e Assinaturas**
Rua da Madalena, 225, 2.º-Dt.º
1100 LISBOA
Telef.: 861520 • 872010 • 877850
Telex: 14557

Marketing
Alberto Câmara Rodrigues

Contabilidade
Serra Brandão, Lda.

Distribuidora
MIDESA — Marco Ibérica,
Distribuição de Edições, S.A.
Rua Principal à Quinta S. João
das Areias, L. 107 • 2685 Sacavém

Produção
Atelier «Software»

Fotolitos
Roseta

Fotocomposição
A.J. - Departamento de Artes Gráficas

Impressão
Printipo — Industrias Gráficas Lda.

Tiragem
10 000 exemplares

Depósito Legal 5599/85

membro da



Associação da Imprensa Não Diária



**PORTE
PAGO**

A Revista SOFTWARE está inscrita no respectivo registo da Direcção Geral de Comunicação Social com o número 110079, e a sua propriedade tem o número 210078. A Revista SOFTWARE é também editada na Bélgica sob o nome de «LE LOGICIEL», na Holanda sob o nome de «DE SOFTWARE» e na Alemanha Ocid. de «SOFTWARE CRIATIV». Todos os direitos reservados. Não é permitida a reprodução total ou parcial das matérias constantes desta edição sem o expresso consentimento da revista e dos autores.

SUMÁRIO

EDITORIAL	1
DESTAQUE	3
CURSOS — PASCAL, BASIC e LING. MAQ.	7 16 36
PROGRAMAS	12
EXCLUSIVO	19
CERTAME	23
ESCAPARATE	31
CONCURSO	34
SOFTWARE CLUBES	38
CLASSIFICADOS e CORREIO	39 46
NOTÍCIAS	47
PASSATEMPO	56
NOVO CURSO — ESTRUTURA DE DADOS	62

editorial

— Artigos de imprensa, declarações de políticos... A escolha pelos CTT/TLP de centrais telefónicas de comutação digital, tornou-se um verdadeiro negócio de Estado. Mas além das pressões existe a técnica...

E quem melhor para vos falar dessa técnica do que a Alcatel Thomson, um dos vencedores do concurso para a instalação das Centrais em Portugal.

Quais as necessidades da rede telefónica nacional, características das centrais digitais, manutenção e benefícios para o assinante, são alguns dos pontos desenvolvidos num esclarecedor artigo que abre este número da Software.

— “Os defensores do LOGO afirmam que as bases fornecidas pelo estudo inicial desta linguagem, são mais apropriadas a um futuro brilhante na programação do que as produzidas pelo BASIC”.

Eis uma das razões que nos levou a querer saber mais sobre esta linguagem, e que o leitor também passará a conhecer melhor.

— Numa fase de euforia a nível de exposições de informática — foi a ENIC, é a INFORPOR em Dezembro e a FILEME em Janeiro/86 — a Software, sempre presente, dá-lhe uma panorâmica da forma como decorreu aquela primeira mostra.

Três temas fortes que escolhemos para lhe “aguçar o apetite”.
Bom trabalho e boa leitura.

Saudações Informáticas



BULL. A ÁRVORE DA COMUNICAÇÃO.

BULL, primeiro construtor europeu de informática e de burótica, propõe às empresas uma oferta coerente nos quatro domínios da informação.

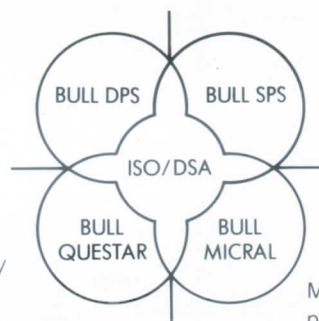
No domínio da informática geral, BULL DPS é uma gama muito vasta de pequenos, médios e grandes sistemas evolutivos cada um numa relação de potência de 1 a 40, cujos logiciais de exploração GCOS garantem a perenidade do vosso investimento em programação.

Para a informática científica e técnica, BULL propõe uma gama de mini e super mini-computadores, sofisticados e potentes, os BULL SPS, particularmente destinados às aplicações da indústria e da investigação.

A informática e a burótica distribuídas beneficiam das novas gerações de terminais e de estações multi-funções polivalentes e ergonómicas, BULL Questar.

Sociedade Publ/ADAP.

Informática geral



Científica e técnica

Informática/burótica distribuídas

Micro-informática profissional

Na micro-informática profissional, BULL Micral representa uma família performante de micro-computadores multipostos ou monopostos de compatibilidade múltipla.

Graças à arquitectura de rede DSA, conforme com as normas internacionais, todos estes sistemas podem comunicar em redes homogéneas ou heterogéneas.





CENTRAIS TELEFÔNICAS DE COMUTAÇÃO DIGITAL

A solução Alcatel Thomson

Artigos de imprensa, declarações de políticos... A escolha pelos CTT/TLP de centrais telefônicas de comutação digital, tornou-se um verdadeiro negócio de Estado. Mas além das pressões, existe a técnica. Foi ela que ditou a decisão dos especialistas a favor, nomeadamente dos equipamentos do Grupo Francês ALCATEL THOMSON.

Decisão justificada? Foi o que tentámos saber.

As centrais concebidas e desenvolvidas pela ALCATEL THOMSON, líder mundial em comutação digital, puseram à prova as suas qualidades, em 49 países do mundo.

Os equipamentos que o Grupo propõe para Portugal são, convém ter presente, os mesmos que foram escolhidos pela Administração dos PTT franceses assim como por 18 países estrangeiros, para equipar as suas redes telefônicas.

Na sua versão "Centros de Trânsito Internacionais", situam-se no primeiro "rang" mundial no que respeita ao número de centros em serviço e encomendados.

UMA RESPOSTA A TODAS AS NECESSIDADES DE UMA REDE TELEFÔNICA NACIONAL

Os sistemas telefônicos digitais propostos para Portugal pela ALCATEL THOMSON, destinam-se às aplicações de trânsito urbano nacional e internacional com operadoras, de assinantes, combinadas e de trânsito com operadoras.

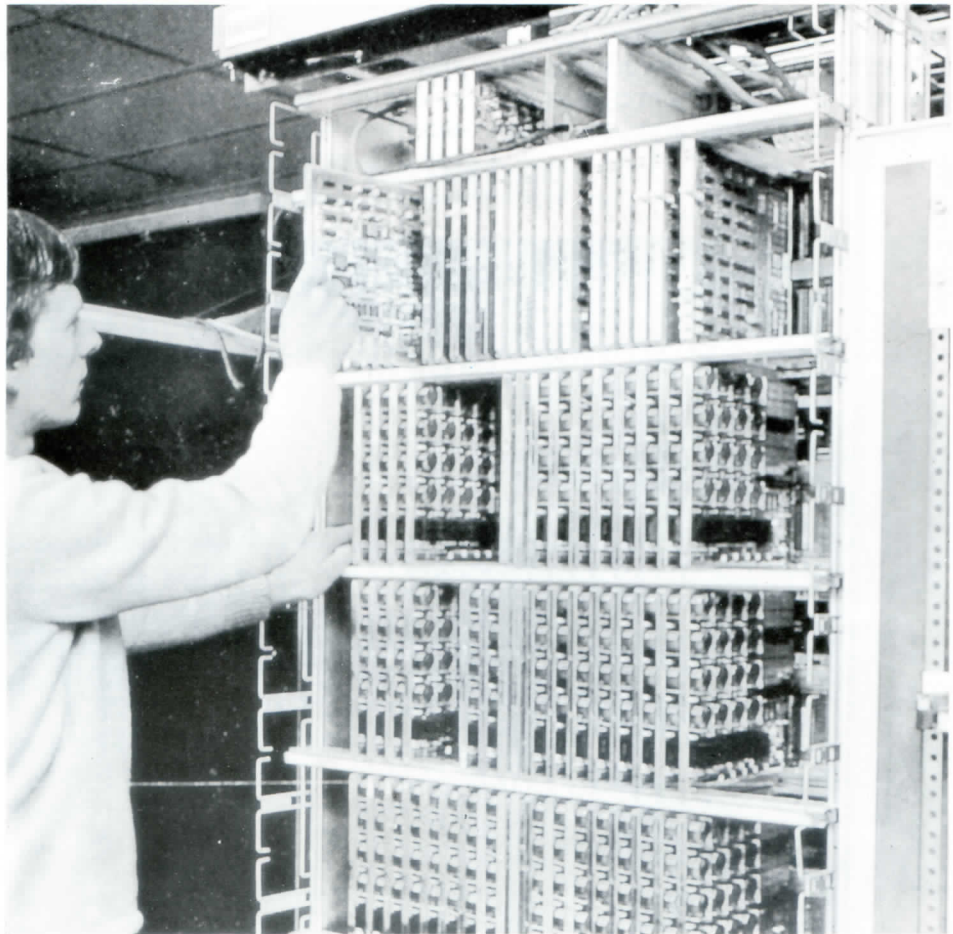
A sua capacidade de escoamento de tráfego é muito superior à dos sistemas baseados nas velhas tecnologias: a sua rede de ligação pode suportar 20.000 comunicações simultâneas com uma probabilidade de bloqueio praticamente nula. O comando por computador (SPC), permite a reconfiguração automática das

vias de transmissão, em função da sua disponibilidade e da quantidade de tráfego. Por outro lado, um sistema de taxação altamente evoluído, com possibilidade de

ra de base.

CARACTERÍSTICAS GERAIS

As centrais telefônicas digitais



Unidade de ligação de assinantes

facturação detalhada, faz com que os sistemas telefônicos de comutação digital se tornem um investimento muito rentável.

Previsto para ser instalado em meio urbano, o autocomutador digital de assinantes ou combinado assinantes/trânsito de grande capacidade da ALCATEL THOMSON escolhido pelos CTT/TLP, permite uma capacidade de ligação até 65.000 assinantes. As zonas periféricas de menor densidade, poderão também ser servidas, uma vez que o referido autocomutador possui uma unidade de ligação distante de assinantes, de que resulta um crescimento do raio de acção e a certeza de rentabilizar em qualquer caso, a infra-estrutu-

ALCATEL THOMSON sobre as quais se fixou a escolha dos CTT/TLP, compreendem quatro tipos de unidades funcionais:

- a rede de ligação (UCX) podendo suportar de 32 a 2048 junções MIC ou seja 65.000 vias telefônicas ou de sinalização, para uma capacidade de escoamento de tráfego de 150 a 20.000 erlangs.

Esta rede é unidireccional (a transmissão das informações telefônicas faz-se sempre no mesmo sentido), dobrada (vias de entrada e de saída ou bidireccionalis ligadas da mesma maneira), dividida em dois ramos independentes



que partilham entre si as chamadas.

A sua velocidade de funcionamento interno é de 4 Mbit/s.

A unidade de comando (UCD) constituída por dois computadores centrais micro-programados, com "palavras" de 32 bits funcionando em regime de partilha de carga, podendo proceder ao tratamento de 350.000 tentativas de chamada, na altura em que se verificar uma sobrecarga de tráfego.

sistema.

Este Logicial é certamente um dos pontos fortes das centrais telefónicas digitais ALCATEL THOMSON. Seguro e altamente potente, apresenta grande possibilidade de evolução graças ao reagrupamento em sub-conjuntos funcionais (SEF), dos programas que dependem unicamente de um número limitado de parâmetros: qualquer evolução tecnológica que se torne necessária para aumentar as suas possibilidades de

controlo passivo e activo, permitem detectar se uma comunicação se estabelece correctamente. Em caso de erro, a unidade de comando duplicada, determina a localização da unidade de segurança em causa e isola-a do tráfego.

Ensaio das junções e das linhas de assinantes: os equipamentos que permitem o ensaio das linhas de assinantes ligadas à central, são comandados a partir de um teleximpressor especializado. Os

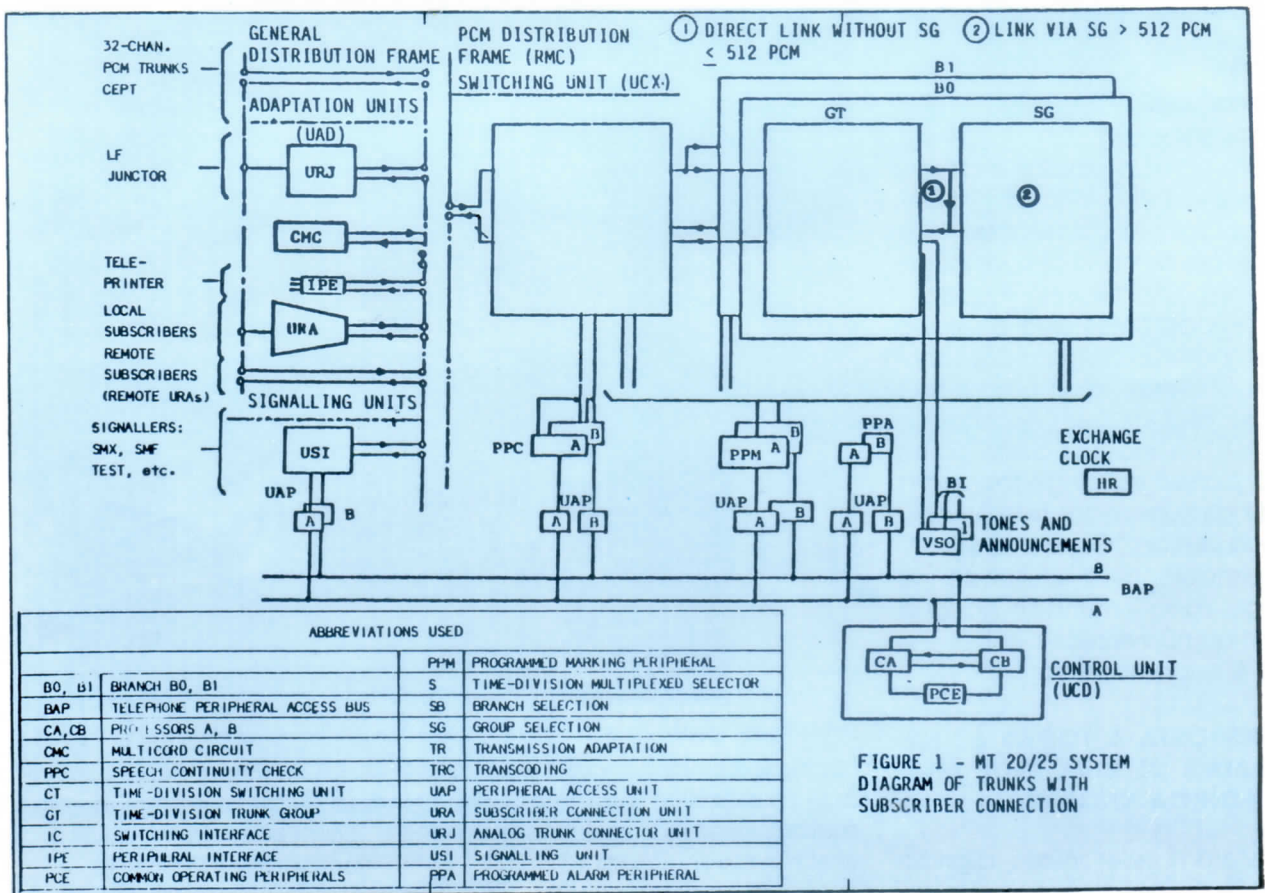


FIGURE 1 - MT 20/25 SYSTEM DIAGRAM OF TRUNKS WITH SUBSCRIBER CONNECTION

A unidade de sinalização (USI) agrupa os órgãos de recepção ou de emissão da sinalização de e para as centrais distantes (sinais de linha e de registadores). Estes sinais são submetidos a um tratamento preliminar através de um microprocessador antes de serem enviados à unidade de comando.

A unidade de adaptação (UAD) constitui a interface entre os acessos analógicos de linhas de assinantes e de circuitos 2 ou 4 fios, e as junções MIC ao "coração" do

aplicação, só influenciará zonas restritas do logicial global.

UMA MANUTENÇÃO SEM HISTÓRIA

As funções de manutenção oferecidas pelas centrais digitais ALCATEL THOMSON escolhidas pelos CTT/TLP, são particularmente interessantes.

Detecção e localização das falhas do equipamento interno na central: os equipamentos de con-

dispositivos de ensaio das junções BF (baixas frequências) permitem ao operador efectuar a supervisão dos juntores, o comando dos ensaios automáticos e os ensaios manuais das junções.

O CONFORTO DO ASSINANTE

O primeiro beneficiário das facilidades oferecidas pelas centrais telefónicas digitais ALCATEL THOMSON será o assinante português, que terá acesso a um grande le-

o mundo num segundo

SERVIÇO PÚBLICO DE COMUNICAÇÃO DE DADOS

Distância não significa tempo.

É necessário chegar a qualquer ponto do globo, em qualquer momento e com a fiabilidade que só um sistema tecnológico evoluído permite.

É preciso comunicar.

E este é o objectivo dos CTT e dos TLP ao criarem o SERVIÇO PÚBLICO DE COMUNICAÇÃO DE DADOS, através da formação do Consórcio Transdata destinado à exploração e comercialização dos novos serviços telemáticos.

Permitindo a troca fácil e eficaz de comunicação — Bancos, Seguros, Turismo, Transportes, Indústria, Administração Pública, Farmácias — terão acesso a bases de dados, tratamento de "stocks", correio electrónico, cálculo científico e industrial.

A rede TELEPAC, associando a economia de exploração à optimização de recursos de transmissão, ligando todo o mundo, transforma o SERVIÇO PÚBLICO DE COMUNICAÇÃO DE DADOS num instrumento indispensável para uma comunicação inteligente.

LISBOA

Av. Fontes Pereira de Melo, 38-9.º

1000 LISBOA

Tel.: 540020 - Telex: 64200

Loja: Forum Picoas

PORTO

Rua de Ceuta, 53-7.º

4000 PORTO

Tel.: 324646 - Telex: 22150



TRANSDATA
CTT e TLP em CONSÓRCIO

SERVIÇO PÚBLICO DE COMUNICAÇÃO DE DADOS



que de serviços, muito deles hoje ainda desconhecidos: telefones com marcador rotativo e de teclas decimais ou MF (frequência CCITT), serviço reduzido, despertar automático, registo de chamada, reenvio temporário, assinante ausente, chamadas de conferência, indicação de chamada com prioridade, linha essencial, facturação detalhada, identificação das chamadas mal intencionadas.

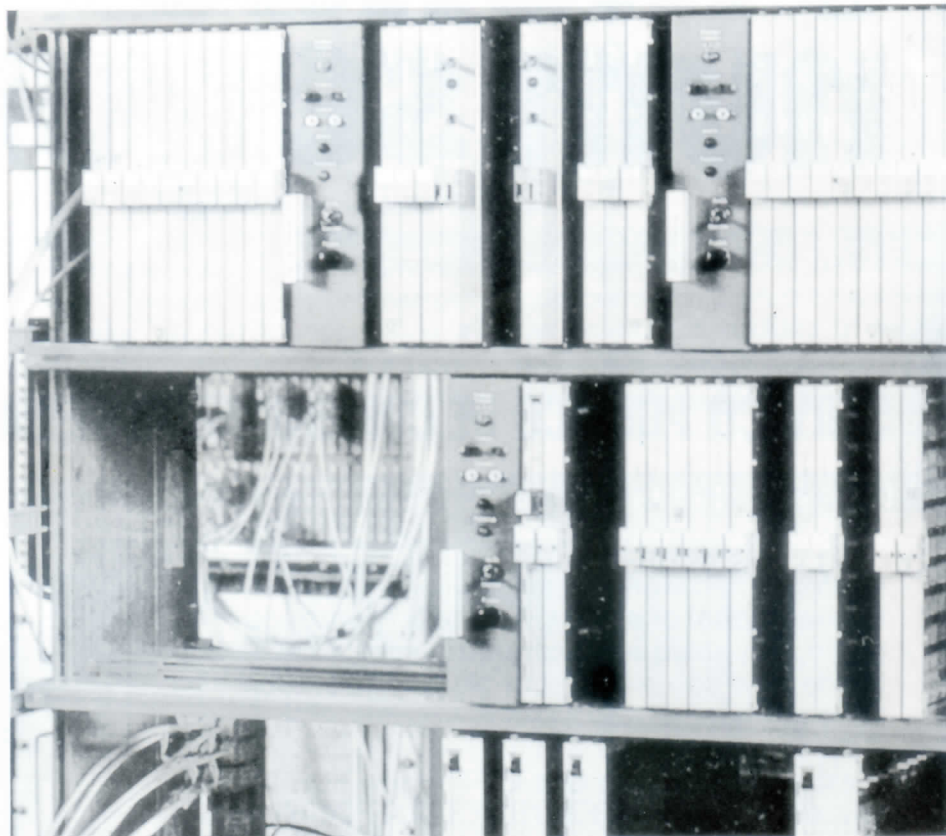
Há ainda a possibilidade de introduzir outras facilidades para os assinantes, isto no caso da Administração dos CTT/TLP o desejar. É ainda necessário sublinhar, que estas centrais podem adaptar-se a todos os planos de numeração e a todos os planos de encaminhamento, oferecendo uma grande possibilidade de escolha de métodos de taxação diferentes, para as chamadas locais, interurbanas e internacionais. Um registo detalhado para cada chamada interurbana ou internacional, poderá ser fornecido aos assinantes que tiverem direito a esse serviço.

Outra vantagem do sistema é o facto do mesmo poder ser integrado em todos os suportes de transmissão estando concebido para tratar os sistemas de sinalização CCITT assim como as suas variantes nacionais.

O FUTURO EM LINHA

A introdução da unidade de ligação de assinantes de terceira geração que permite a ligação digital a 64 Kbit/s e a utilização do código de sinalização N.º 7, para o qual as centrais telefónicas digitais ALCATEL THOMSON estão já preparadas, permitir-lhes-à responder a todas as necessidades da Rede Digital com Integração de Serviços (transmissão simultânea numa linha telefónica, da voz, dos dados, do texto, etc...)

A França, pioneiro mundial neste domínio, utiliza evidentemente os equipamentos desenvolvidos pela ALCATEL THOMSON para as experiências em grandeza real

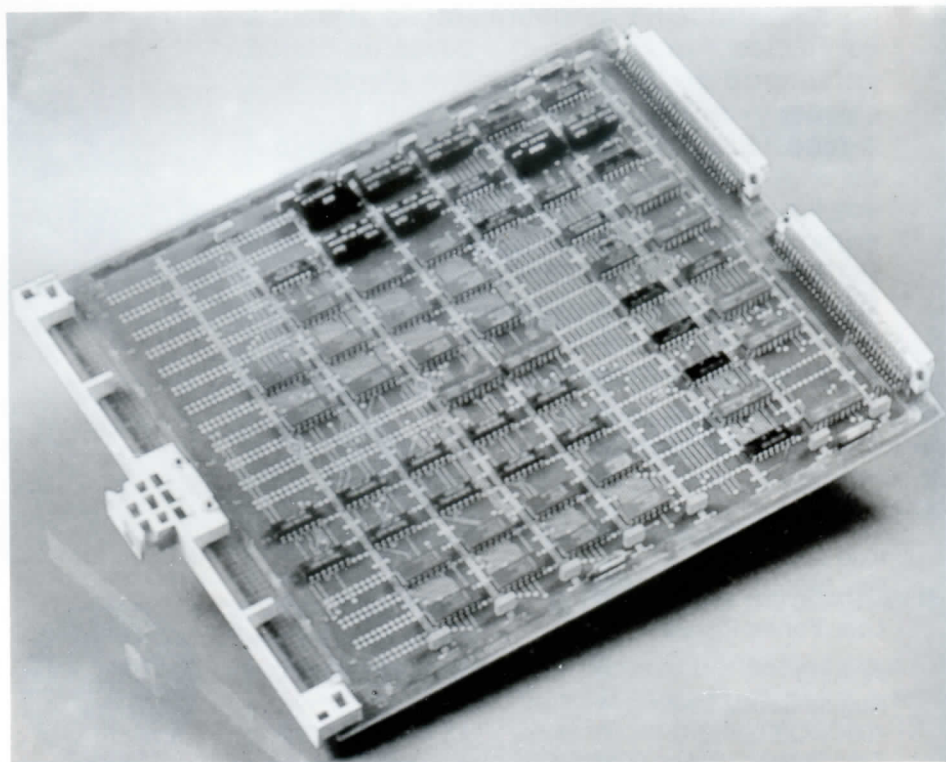


Unidade de sinalização

que está a realizar actualmente, na sua Rede Telefónica Digital.

Com a comutação digital e as

centrais digitais ALCATEL THOMSON, os CTT/TLP, vão "colocar o futuro em linha". □



Filtro digital



PASCAL

FERNANDO GARCIA

Introdução

Um dos métodos de desenvolvimento de algoritmos mais utilizado é sem dúvida a análise *top-down* (traduzido à letra: de cima para baixo; mais correctamente: do geral para o particular).

Este método consiste em ir refinando progressivamente uma solução para um problema em níveis sucessivos de particularização. O desenho do algoritmo começa pela sua divisão em vários blocos, a definição da função de cada um dos blocos e a definição da forma como deve ser encadeada a sua execução. Cada bloco constitui uma unidade estanque podendo a ele aplicar-se o mesmo processo global: dividi-lo em sub-blocos. Este processo vai-se aplicando sucessivamente até se chegar a um nível em que os blocos apresentam um conjunto de operações que, dada a sua simplicidade, não necessitam de ser decompostas em outras intermédias. Assim os blocos dos níveis superiores ocupam-se com os problemas de um forma geral, enquanto que os blocos dos níveis inferiores lidam com os detalhes. Nos blocos intermédios assiste-se à passagem progressiva de uma abordagem global para uma preocupação crescente com os pormenores.

Este modo de decompor um problema está directamente ligado a nossa incapacidade de abarcar simultaneamente uma grande quantidade de informação. Por esta razão os nossos raciocínios são estruturados em níveis.

Ora, a ideia de uma linguagem de programação evoluída é facilitar a comunicação entre o Homem e o computador o mais próxima possível da forma como usualmente estrutura os seus raciocínios. Por isso o Pascal apresenta uma hierarquização por meio de uma estratificação em níveis.

Assim, um programa em Pascal é dividido em vários blocos designados por procedimentos (*Procedu-*

re). Um procedimento não é mais do que um conjunto de instruções que, ao ser executadas produzem determinadas acções tal e qual como os programas que temos visto neste curso.

A cada procedimento damos um nome que deve caracterizar a acção que desempenha (tal como atribuímos nomes sugestivos aos programas que fizemos). Sempre que no programa é necessário executar essa acção em vez de incluirmos as instruções necessárias, basta especificar o nome do procedimento. Assim o programa principal apenas exprime em termos gerais a sequência de acções a realizar e não os pormenores dessas acções; esses são deixados para os procedimentos.

Procedimentos

Nesta secção é apresentado o desenvolvimento de um programa, recorrendo à sua divisão em procedimentos. Nela o leitor poderá encontrar a concretização das ideias apresentadas na introdução. Essas ideias provavelmente afiguraram-se-lhe um pouco aridas embora eu pense serem necessárias para encarar de uma forma natural o aparecimento dos procedimentos em Pascal.

O problema aqui abordado pode ser solucionado por meio de um algoritmo muito utilizado quando não se conseguem encontrar outros mais eficientes (pode parecer um pouco contraditório mas muitas vezes não se conseguem arranjar bons algoritmos para a resolução de certos problemas).

O problema a resolver é o seguinte (*):

Gerar uma sequência de n caracteres de um conjunto com 3 caracteres A, B e C tal que não contenha nenhum par de subseqüências adjacentes iguais.

Por exemplo, supondo $n=6$, as seqüências ABCABA e ABCBAC

são válidas enquanto que as seqüências AABCAB (tem dois A's seguidos) e ABCABC não são válidas.

O algoritmo utilizado consiste em gerar sistemáticamente uma seqüência de caracteres, acrescentando caracter a caracter, e testá-la para verificar se obedece às condições especificadas, sendo modificada no caso de não obedecer.

O processo inicia-se com uma seqüência vazia. O passo seguinte consiste em *acrescentar*, um a um, caracteres à seqüência. Cada vez que um caracter é acrescentado há que *verificar* se esta obedece às restrições impostas. No caso afirmativo a seqüência até ao momento é *válida* e pode de novo ser acrescentada. Caso contrário é considerada não válida e tem que se *modificar*. O algoritmo termina quando o comprimento de uma seqüência válida for igual a n .

Portanto as condições iniciais são:

$sequencia[comprimento] := succ(sequencia[comprimento])$

A segunda é um pequeno artifício útil na simplificação do algoritmo.

O corpo do algoritmo é:

REPETIR

SE *válida* ENTÃO *acrescentar*

SENÃO *modificar*

ATÉ *comprimento igual a n* E *válida*

No final há que imprimir a seqüência encontrada:

Imprimir

Nesta primeira aproximação não sabemos ainda como vamos proceder para *acrescentar*, *modificar* e *verificar* a seqüência, assim como para a *imprimir*. No entanto, se consideramos que estas acções são efectuadas por procedimentos, exactamente com aqueles nomes, podemos imediata-



cont. PASCAL

```
comprimento := 0;
valida := true;
REPEAT
  IF valida THEN acrescentar
    ELSE modificar;
  verificar
UNTIL (comprimento = n) AND valida;
```

Neste conjunto de instruções aparece uma coisa nova: uma variável lógica — *válida*. Uma variável lógica pode assumir apenas dois valores: falso (false) ou verdadeiro (true) — ou sim ou não, ou 0 ou 1,...

Assim, e no caso presente, a variável *válida* indica apenas uma de duas coisas: a sequência ou é válida (*válida = true*) ou não é válida (*válida = false*).

A lógica a dois valores foi estruturada por um matemático de nome Boole e por isso é corrente apelidá-la de booleana. Dá a declaração em Pascal:

```
valida : boolean;
```

A sequência de caracteres é uma lista, de caracteres, cuja dimensão, *n*, é conhecida à partida. Em Pascal pode ser representada por um *array* de caracteres:

```
sequencia : ARRAY [1..n] OF char;
```

Até este momento definimos a forma como representar os dados e a estrutura geral do processamento. Estamos pois na posse dos elementos necessários para escrever a parte do programa de declaração do dados e o programa principal:

```
PROGRAM sequencias (output);
```

```
CONST
  n = 6;
```

```
VAR
  sequencia : ARRAY [1..n] OF char;
  comprimento : 0..n;
  valida : boolean;
```

```
BEGIN
  comprimento := 0;
  valida := true;
  REPEAT
    IF valida THEN acrescentar
      ELSE modificar;
    verificar
  UNTIL (comprimento = n) AND valida;
  imprimir
END.
```

Passando à fase seguinte vamos desenvolver os procedimentos.

A acção de *acrescentar* a sequência é muito simples, é apenas aumentar de um o seu comprimento:

```
comprimento := comprimento + 1;
```

e escolher o novo carácter para estender a sequência. Para sistematizar a geração da sequência o carácter escolhido é sempre o primeiro do conjunto dos possíveis:

```
sequencia[comprimento] := 'A'
```

O procedimento *acrescentar* é pois:

```
PROCEDURE acrescentar;
```

```
BEGIN
  comprimento := comprimento + 1;
  sequencia[comprimento] := 'A'
END
```

O *modificar* a sequência é efectuado quando o último carácter acrescentado conduz a uma sequência não válida. Portanto, consistirá em trocar esse carácter por outro, dos possíveis, com o intuito de verificar se este já conduz a uma sequência correcta. Também para sistematização troca-se sempre pelo carácter seguinte, ou seja, pelo seu sucessor no conjunto dos caracteres possíveis:

```
sequencia[comprimento] :=
  succ(sequencia[comprimento])
```

A função *succ()* (*successor*) dá o sucessor, isto é, o carácter que se encontra imediatamente a seguir, segundo uma ordem pré-definida — numérica para os carácter

de '0' a '9' e alfabética para os caracteres de 'A' a 'Z'.

Claro que quando se chega ao último carácter do conjunto, 'C', temos que parar. Neste caso a conclusão é esta: se, qualquer que seja o carácter final experimentado, a sequência não é válida, então há que retroceder, ir ao carácter anterior, trocá-lo por outro e ver se assim se chega já a uma sequência válida. Mas pode ainda acontecer que o carácter anterior seja também o último. Ora, enquanto isso acontecer tem que se retroceder:

```
WHILE sequencia[comprimento] = 'C'DO
  comprimento := comprimento - 1;
```

O procedimento *modificar* é portanto:

```
PROCEDURE modificar;
```

```
BEGIN
  WHILE sequencia[comprimento] = 'C'DO
    comprimento := comprimento - 1;
    sequencia[comprimento] :=
      succ(sequencia[comprimento])
END
```

Convém aqui notar que o método sistemático como se vai gerando a sequência assegura que não há nenhum arranjo possível dos símbolos que seja esquecido. Por outro lado não se limita a gerar todos os arranjos possíveis, pelo contrário, actua de uma forma inteligente: sempre que se conclui que um certo símbolo não pode ocupar uma determinada posição, porque não conduz a uma solução, não se testam mais nenhuma sequência com esse símbolo nessa posição, troca-se imediatamente pelo símbolo seguinte.

Para que o leitor possa verificar que realmente é assim proponho-lhe que faça o papel do processador, para o programa em questão, durante algumas iterações escrevendo as sub-sequências que vai gerando.

A verificação da validade da sequência foi deixada inteiramente para o procedimento *verificar*.



cont. PASCAL

Dada esta separação, derivada do modo de desenvolvimento do algoritmo, é possível por simples substituição deste procedimento gerar outras sequências com outro tipo de restrições que as atrás enunciadas. Até ao momento o programa possui um caracter bastante geral.

O procedimento *verificar* tem que ver se a sequência contém algum par de subsequências adjacentes iguais. Para isso tem que comparar todos os pares de sub-sequência adjacentes de comprimento igual a 1 (não pode haver dois caracteres iguais seguidos) 2, 3, ..., até $n/2$ (as subsequências adjacentes mais extensas são as duas metades da sequência).

Note-se que se encontram duas subsequências adjacentes iguais não vale a pena continuar a comparação, a sequência que as contém pode ser imediatamente dada como não válida.

Esquecendo por enquanto como se vão comparar as duas subsequências temos qualquer coisa como:

```
valida := true;
subcomp := 0;
metade := comprimento div 2;
WHILE valida AND (subcomp < metade) DO
  BEGIN
    subcomp := subcomp + 1;
    valida := "Todas as subsequências adjacentes
              de comprimento subcomp são diferentes"
  END
```

Como é evidente *metade* é igual à divisão inteira $n/2$, o que em Pascal é indicado por:

```
metade := comprimento div 2;
```

Notemos ainda que, quando se acrescenta um caracter à sequência, se sabe à partida que ela até ao momento era válida. Por isso não vale a pena comparar todas as subsequências adjacentes de determinado comprimento, basta comparar as que contém o caracter acrescentado. Assim, para cada comprimento

há só a comparação de um par de subsequências.

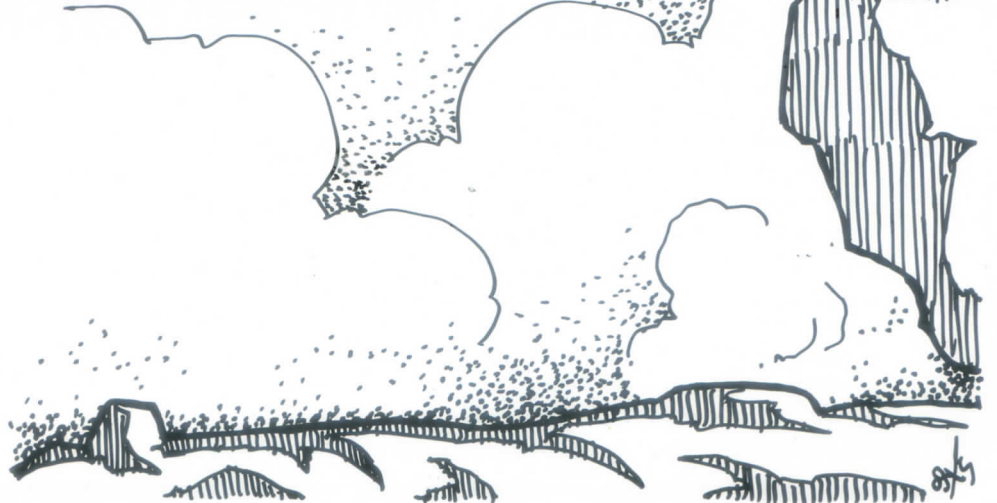
A comparação de duas subsequências adjacentes de comprimento *subcomp* pode ser feita através de um ciclo que compara por ordem os caracteres que as compõe. Para que as subsequências sejam diferentes basta existir um par de caracteres diferentes. Se tal acontecer não vale a pena continuar a comparar os próximos caracteres e, nestas condições, o ciclo deve parar antes de chegar ao fim das subsequências.

Traduzindo este raciocínio temos:

```
ordem := 0;
REPEAT
  valida := sequencia[comprimento-ordem] =
            sequencia[comprimento-subcomp-ordem];
  ordem := ordem + 1
UNTIL valida OR (ordem = subcomp)
```

Os índices do *array* podem parecer um pouco complicados. Contudo, se o leitor reflectir um

pouco, conclui rapidamente que conduzem à comparação dos caracteres da subsequência (de comprimento *subcomp*) no fim da sequência com a ajacente.





HICOLUX

© BURROUGHS

A3 o sistema de hoje a pensar no futuro

BURROUGHS um dos construtores líderes mundiais de grandes sistemas informáticos apresenta o A3 da performante série A.

Instale um centro de processamento rentável, produtivo e adaptado às exigências modernas de uma produção racional e eficiente da informática.

A3 apresenta uma liderança tecnológica em Hardware e Software.

- Memória central 48 MB — Nova tecnologia em Discos Fixos.
- 126 Canais de comunicações — Impressora Laser
- Processadores Independentes para Periféricos, Telecomunicações e Manutenção
- INTERPRO — O conceito mais evoluído em Sistemas Operativos para grandes sistemas de computadores.
- LINC — Linguagem de Desenvolvimento da Quarta Geração.

Um máximo de tecnologia e produtividade para a nova informática nacional.

 **Burroughs**
A chave de cada sector chave



cont. PASCAL

Deste modo chegamos ao procedimento *verificar*:

```
PROCEDURE verificar;

BEGIN
  valida := true;
  subcomp := 0;
  metade := comprimento div 2;
  WHILE valida AND (subcomp < metade) DO
  BEGIN
    subcomp := subcomp + 1;
    ordem := 0;
    REPEAT
      valida := sequencia[comprimento-ordem] =
        sequencia[comprimento-subcomp-ordem];
      ordem := ordem + 1
    UNTIL valida OR (ordem = subcomp)
  END
END;
```

O procedimento *imprimir*, dada a sua simplicidade não é tratado em pormenor.

Juntando tudo temos o programa:



Aconselho ainda o leitor a reler a secção anterior, depois de compreender e por funcionar este programa, a fim de consolidar as noções nela apresentadas. A boa ou má estruturação de um programa determina drasticamente a sua compreensão, adaptação a outras tarefas semelhantes aquela a que inicialmente era destinado, funcionalidade, transportabilidade (poder por-se a funcionar noutro computador que não aquele onde foi feito),..., enfim a utilidade do programa.

Exercício:

Modifique o programa de modo a gerar, não apenas uma, mas todas as sequências de comprimento *n* que não contenham subseqüências adjacentes iguais.

Dada a boa estruturação do programa esta tarefa é relativamente simples. □

(*) in Niklaus Wirth, Programação sistemática em Pascal

```
PROGRAM sequencias (output);

CONST
  n = 6;

VAR
  sequencia : ARRAY [1..n] OF char;
  comprimento, ordem, subcomp, posicao : 0..n;
  metade : 1..n;
  valida : boolean;

PROCEDURE acrescentar;

BEGIN
  comprimento := comprimento + 1;
  sequencia[comprimento] := 'A'
END;

PROCEDURE modificar;

BEGIN
  WHILE sequencia[comprimento] = 'C' DO
    comprimento := comprimento - 1;
    sequencia[comprimento] := succ(sequencia[comprimento])
  END;

PROCEDURE verificar;

BEGIN
  valida := true;
  subcomp := 0;
  metade := comprimento div 2;
  WHILE valida AND (subcomp < metade) DO
  BEGIN
    subcomp := subcomp + 1;
    ordem := 0;
    REPEAT
      valida := sequencia[comprimento-ordem] =
        sequencia[comprimento-subcomp-ordem];
      ordem := ordem + 1
    UNTIL valida OR (ordem = subcomp)
  END
END;

PROCEDURE imprimir;

BEGIN
  FOR posicao := 1 TO n DO
    write (sequencia[posicao])
  END;

BEGIN {sequencias}
  comprimento := 0;
  valida := true;
  REPEAT
    IF valida THEN acrescentar
      ELSE modificar;
    verificar
  UNTIL (comprimento = n) AND valida;
  imprimir
END.
```



FICHEIROS

ANTÓNIO TAVARES

O programa que apresento, elabora pequenos ficheiros que podem ser relativos a pessoas ou a quaisquer outras matérias e são ordenados alfabeticamente a partir do nome.

A cada nome poderá fazer corresponder um pequeno conjunto de dados de identificação que, segundo a estrutura do programa, não deverá exceder os 192 caracteres (6 linhas); por sua vez, o nome não deverá exceder os 25 caracteres.

Em principio, o número de fichas ou registos não deverá exceder os 150 com vista a salvaguardar alguma RAM livre.

As dimensões do ficheiro são definidas na linha n.º 1084, onde N é a variável indexada que contém os nomes e D é a que contém os dados.

Se, por qualquer motivo, desejar aumentar o n.º de caracteres (ou capacidade) da variável que contém os dados, terá que a redimensionar, diminuindo o n.º de registos e aumentando o n.º de caracteres em D.

Se, por outro lado, desejar aumentar o n.º de fichas terá que diminuir o n.º de caracteres atribuídos aos dados.

Estes ficheiros podem ser guardados em cassette de duas maneiras:

1) Gravados, independentemente do programa como "arrays".

2) Gravados juntamente com o programa.

Estas operações são efectuadas a partir das opções fornecidas com o programa.

Se desejar atribuir os seus próprios nomes ao programa ou aos "arrays" a gravar, proceda do seguinte modo:

1) Gravar somente os ficheiros: Edite as linhas n.º 500, 510, 670 e 680 e atribua os nomes pretendidos.

2) Gravar juntamente com o programa:

Edite a linha n.º 630 e proceda como anteriormente.

Pode também, se assim o desejar, gravar somente o programa (sem os ficheiros).

Notas:

Se, ao emendar os dados (opção 7), tiver que apagar espaços (tecla E) e o último carácter da 6.ª linha de dados aparecer repetido após ter realizado a operação, não se preocupe, pois isso desaparece quando se faz a listagem normal (opção 4).

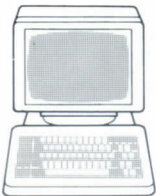
A opção 2 é bivalente: permite introduzir uma lista a partir da cassette com listagem posterior dos nomes, ou então listá-los a partir da RAM quando a lista já está presente.

ATENÇÃO: A opção 1 e a opção 2 via cassette apagam qualquer ficheiro presente em memória (bem como o comando 'RUN').

Notará que, depois de ter introduzido os dados relativos a um nome, aparece um par de aspas vazias (espera de introdução alfanumérica). Se desejar prosseguir com a introdução de ficheiros responda a estas aspas carregando somente em 'enter'. Se desejar interromper a introdução de ficheiros responda às aspas com a letra F e, depois, 'ENTER'.

Se desejar copiar para a impressora quaisquer dados ou lista, responda à indicação no fim do ecran (pulse 'ENTER') com 'SHIFT+BREAK' e execute o comando 'COPY'. Depois deste comando ter sido executado, introduza 'GO TO 10' para voltar à lista de opções.

NUNCA utilize o comando 'RUN', pois apagaria o ficheiro em memória.



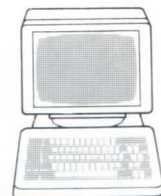
SPECTRUM
2068
2048

```

1 REM Organização de ficheiros
3 REM © Por Antonio M.Freire
  Ribeiro Tavares
6 REM Nomes -Max.25 caracteres
  dados - 192
7 REM
8 REM Listagem das opções
10 CLS : PRINT INVERSE 1;"ORGA
NIZACAO DE FICHEIROS";AT 2,3;"Te
cla";AT 2,13;"Opcao": PRINT AT 4
,5;"0";AT 4,13;"*Fim de opcao";A
T 5,5;"1";AT 5,13;"*Introducao d
e uma";AT 6,13;"Lista (via tecla
do)";AT 7,5;"2";AT 7,13;"*Carreg
ar lista e/";AT 8,13;"ou listar
nomes";AT 9,5;"3";AT 9,13;"*Adic
ao de nomes a";AT 10,13;"Uma lis
ta ordenada"
20 PRINT AT 11,5;"4";AT 11,13;
"*Pesquisa (nomes e";AT 12,13;"d
ados)";AT 13,5;"5";AT 13,13;"*Gr
avar lista orde-";AT 14,13;"nada
";AT 15,5;"6";AT 15,13;"*Apagar
nome";AT 16,5;"7";AT 16,13;"*Eme
ndar dados";AT 17,5;"8";AT 17,13
;"*Gravar programa";AT 18,13;"co
m a lista";AT 19,5;"9";AT 19,13;
"*Gravar programa"
22 REM Seleccao de opcoes
24 POKE 23609,50
30 INPUT "Opcao: ";OP
40 IF OP=0 THEN GO TO 1090
50 IF OP=1 THEN CLS : GO SUB 1
084: LET L1=1: INPUT "Numero de
pessoas (ou nomes)=";N: GO TO 150
60 IF OP=2 THEN CLS : GO SUB 4
50: LET N=F: GO TO 382
70 IF OP=3 THEN CLS : INPUT "N
um.de pessoas a adicionar=";PA:
LET N=F+PA: LET L1=F+1: GO TO 15
0
80 IF OP=4 THEN DIM E$(25): CL
S : LET N=F: GO TO 530
90 IF OP=5 THEN CLS : GO TO 62
0
100 IF OP=6 THEN DIM E$(25): CL
S : GO TO 710
110 IF OP=7 THEN CLS : DIM E$(2
5): GO SUB 850: GO TO 940
130 IF OP=8 OR OP=9 THEN GO TO
620
142 REM Entrada de nomes
e dados
150 FOR M=L1 TO N
160 PRINT FLASH 1;AT 0,0;M
170 LET F=M
    
```



cont. FICHEIROS



SPECTRUM

2068

2048

```

180 PRINT INVERSE 1; AT 21,0; "<-
---25 caracteres--->"
190 PRINT FLASH 1; AT 21,24; "I"
200 INPUT LINE N$(M)
210 BORDER 0; PRINT AT 0,3; N$(M)
) : PRINT INVERSE 1; AT 1,0; "Entra
da de dados (max.6 linhas)": IN
PUT AT 20,0; AT 2,0; LINE D$(M); B
$: CLS : BORDER 7: IF B$="F" OR
B$="r" THEN GO TO 240
220 NEXT M
230 LET N=F
240 CLS
242 PRINT FLASH 1; AT 10,10; "ORD
ENANDO!"
250 LET L=1
252 REM Ordenacao alfabetica
260 FOR I=1 TO 25
270 FOR K=L TO N-1
280 FOR J=K+1 TO N
290 IF I=1 THEN GO TO 310
300 IF N$(K,1 TO I-1) < N$(J,1
TO I-1) THEN GO TO 370
310 IF CODE N$(K,I) <= CODE N$(
J,I) THEN GO TO 350
320 LET E$=N$(K); LET F$=D$(K
)
330 LET N$(K)=N$(J); LET D$(K
)=D$(J)
340 LET N$(J)=E$; LET D$(J)=F
$
350 NEXT J
360 NEXT K
370 NEXT I
380 IF K < N-1 THEN LET L=K+1: GO
TO 260
382 CLS
384 REM Listagem dos nomes
390 FOR M=1 TO N
410 PRINT N$(M)
420 NEXT M
430 PRINT #0; "Num.de pessoas (ou
nomes) ="; F
440 PRINT #0; FLASH 1; "Pulse 'E
NTER'": PAUSE 0: GO TO 10
442 REM Rot.opcao 2
450 INPUT "Lista em RAM ou fita
? (R/F)": R$: IF R$="R" OR R$="r"
THEN RETURN
452 REM Carregar lista a partir
de fita
454 GO SUB 1084
460 INPUT "Numero de pessoas (ou
nomes) em fita="; F
480 LET N=F
490 PRINT #0; FLASH 1; "Prepare
o gravador e pulse 'ENTER'": PAU
SE 0: CLS
500 LOAD "nome" DATA N$(0)
510 LOAD "dados" DATA D$(0)
520 RETURN
522 REM Pesquisa de dados
530 PRINT FLASH 1; "Introduza o
nome"
540 INPUT LINE A$: IF LEN A$=25
THEN LET C$=A$: CLS : GO TO 560
550 LET C$=A$+E$(1 TO (25-LEN A
$))
560 FOR M=1 TO N
570 IF C$=N$(M) THEN PRINT INU
ERSE 1; AT 0,0; N$(M); PRINT : P
RINT : PRINT INVERSE 1; "Dados": P
RINT : PRINT D$(M); PRINT #0; FL
ASH 1; "Pulse 'ENTER'": PAUSE 0:
GO TO 10
580 NEXT M
590 CLS
600 PRINT FLASH 1; AT 0,0; "Nao e
xiste": PRINT #0; FLASH 1; "Pulse
'ENTER'": PAUSE 0: GO TO 10
612 REM Rotina de gravacao
620 PRINT #0; FLASH 1; "Prepare
o gravador e pulse 'ENTER'": PAU
SE 0: CLS
630 IF OP=8 THEN SAVE "FICH+" L
INE 10: GO TO 1090
640 IF OP=9 THEN CLEAR : SAVE "
FICH-" LINE 10: GO TO 1090

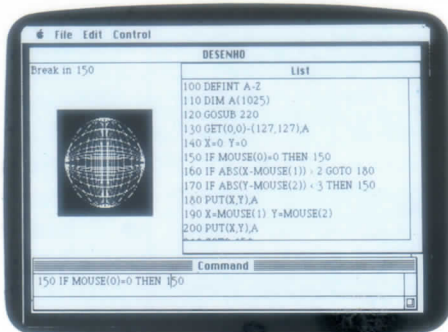
```

```

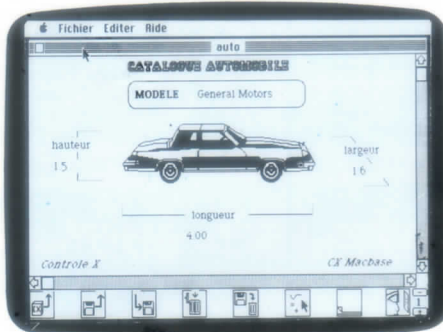
670 SAVE "nome" DATA N$(0)
680 SAVE "dados" DATA D$(0)
690 INPUT FLASH 1; "Deseja verif
icar as gravacoes?(S/N)": K$: IF
K$="S" OR K$="s" THEN PRINT #0;
FLASH 1; "Prepare o gravador e pu
lse 'ENTER'": PAUSE 0: CLS : VER
IFY "" DATA N$(0): VERIFY "" DATA
D$(0)
700 GO TO 10
702 REM Apagar nome
710 PRINT INVERSE 1; "Escreva no
me a apagar": LET N=F
720 INPUT LINE A$: IF LEN A$=25
THEN LET C$=A$: CLS : GO TO 740
730 LET C$=A$+E$(1 TO (25-LEN A
$))
740 FOR M=1 TO N
750 IF C$=N$(M) THEN LET L2=M:
GO TO 790
760 NEXT M
770 CLS
780 PRINT FLASH 1; AT 0,0; "Nao e
xiste": PRINT #0; FLASH 1; "Pulse
'ENTER'": PAUSE 0: GO TO 10
790 FOR K=L2 TO F
800 LET N$(K)=N$(K+1): LET D$(
K)=D$(K+1)
810 NEXT K
820 LET F=F-1: LET N=F
830 INPUT "Prende apagar outr
o nome?(S/N)": S$: IF S$="S" OR S
$="s" THEN CLS : GO TO 710
840 LET F=N: GO TO 10
842 REM Emendar dados
850 PRINT FLASH 1; "Introduza o
nome"
860 INPUT LINE A$: IF LEN A$=25
THEN LET C$=A$: CLS : GO TO 880
870 LET C$=A$+E$(1 TO (25-LEN A
$))
880 FOR M=1 TO N
890 IF C$=N$(M) THEN PRINT INU
ERSE 1; AT 0,0; N$(M): RETURN
900 NEXT M
910 CLS
920 PRINT FLASH 1; AT 0,0; "Nao e
xiste": PRINT #0; FLASH 1; "Pulse
'ENTER'": PAUSE 0: GO TO 10
940 LET AA=1: LET LL=LEN D$(M):
LET Z$=D$(M,1)
950 PRINT AT 2,0; D$(M): LET D$(
M,AA)=Z$
960 PRINT AT 11,0; "Deslocar o c
ursor: <-5 0->"; AT 13,0; "Emenda
r/Inserir: A"; AT 15,0; "Apa
gar espaço: E"; AT 17,0; "
Opcoes: 1"
970 IF INKEY$="8" AND AA < LL THE
N LET AA=AA+1
980 IF INKEY$="5" AND AA > 1 THEN
LET AA=AA-1
990 IF INKEY$="1" THEN GO TO 10
1000 IF INKEY$="A" OR INKEY$="a"
THEN GO SUB 1040
1010 IF INKEY$="E" OR INKEY$="e"
THEN GO SUB 1070
1020 PRINT AT 19,12; "Caractere:"
; AT 21,0; "D$("; M; ", "; AA; ")"; AT 2
1,15; D$(M,AA)
1030 LET Z$=D$(M,AA): LET D$(M,A
A)=" "; GO TO 950
1040 INPUT FLASH 1; BRIGHT 1; "Es
creva caractere:"; H$
1050 LET D$(M,AA)=H$
1060 RETURN
1070 INPUT ; FLASH 1; BRIGHT 1; "
Pulse 'ENTER'": LINE U$: LET D$(
M,AA)=CHR$ 9
1080 RETURN
1082 REM Dimensionamento dos
ficheiros
1084 DIM N$(150,25): DIM D$(150,
192)
1086 RETURN
1090 REM FIM

```

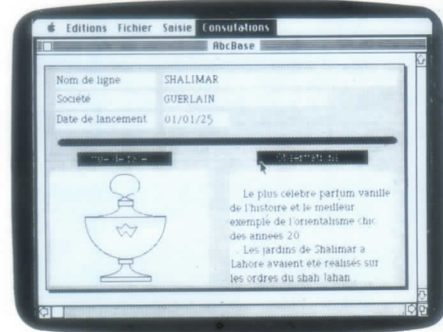
Interrompemos esta revista para an



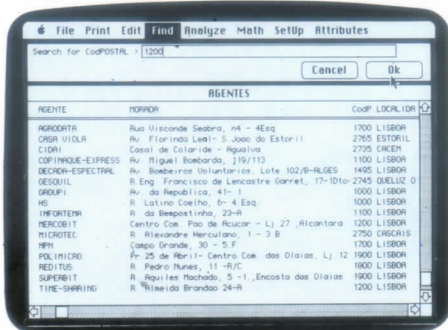
Microsoft BASIC
Linguagem de Programação



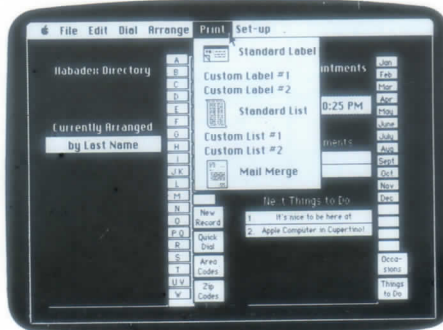
CX MacBase
Gestão de Ficheiros



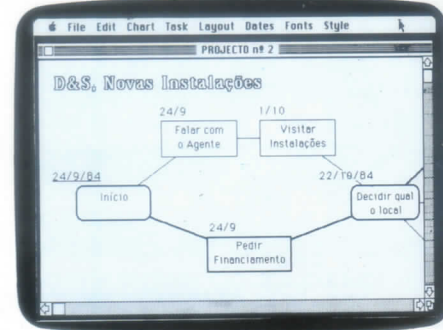
ABC Base
Gestão de Ficheiros



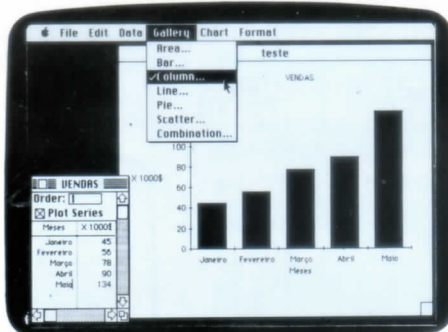
Over VUE
Gestão de Ficheiros



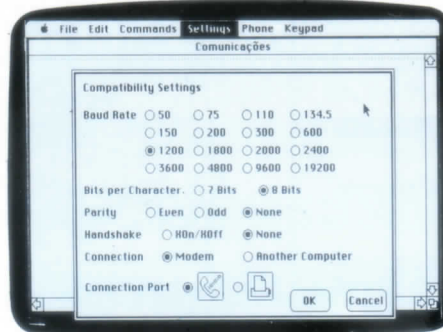
Habadex
Agenda



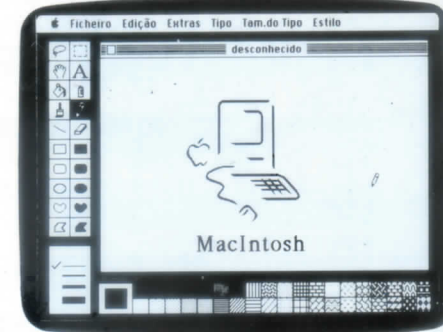
Mac Project
Controlo de Projectos



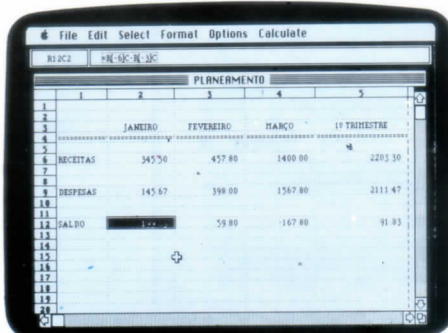
Microsoft CHART
Gráficos



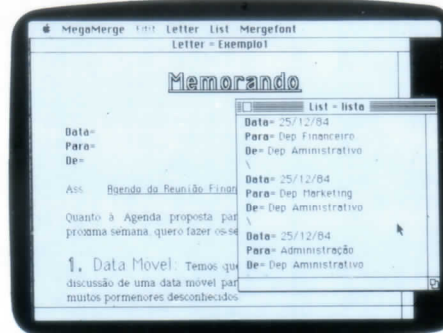
Mac Terminal
Comunicações



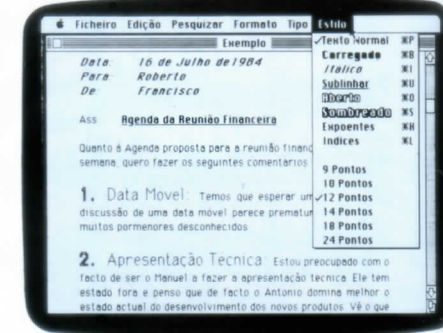
Mac Paint
Tratamento de Desenhos



Multiplan
Folha de Cálculo Electrónica



MegaMerge
Endereçamento Automático



MacWrite
Processamento de texto

As aplicações acima referidas são alguns exemplos do que se encontra desenvolvido para o Macintosh. Para mais informações, contacte MICROPOR — Sistemas de Informática, S.A.R.L — Av. das Túlipas, Lote 11, r/c Dr.º — MIRAFLORES — 1495 LISBOA, Tel. 2105295, 2105296, 2105297, Telex 13248 MICROP P, ou qualquer dos Agentes Apple autorizados.

Annunciar alguns programas importantes.

Todos os dias, novas aplicações estão sendo desenvolvidas para o computador pessoal Macintosh™.

Aplicações de tratamento de texto, cálculo electrónico, gráficos, controlo de projectos, bases de dados, comunicações ...

E também aplicações que permitem ao Macintosh cumprir tarefas que até agora nunca foram possíveis de executar no outro computador.

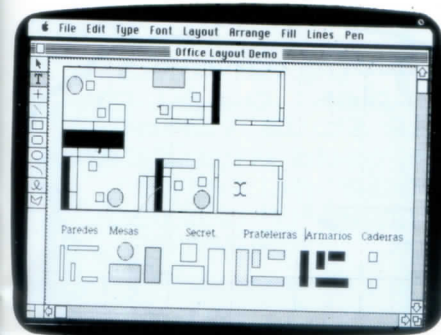
O que significa que o Macintosh se está a afirmar como o computador pessoal mais famoso internacionalmente.

Qualquer Agente autorizado Apple poderá demonstrar-lhe facilmente este facto.

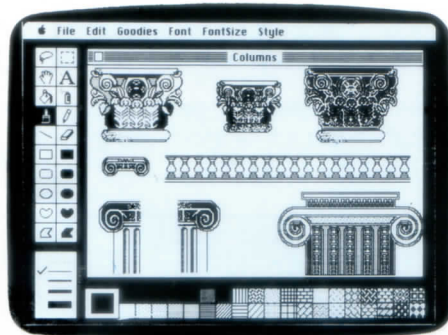
Peça-lhe para ver o computador que possui aplicações compatíveis com os seres humanos.



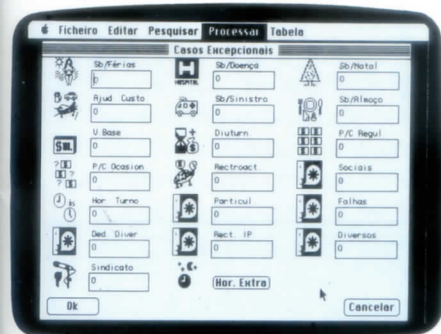
®



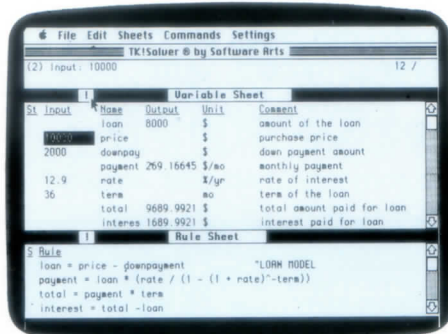
MacDraw
Arquitectura



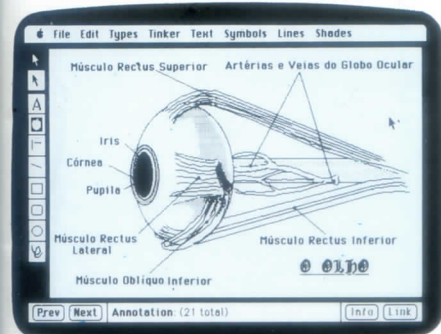
Da Vinci
Motivos Decorativos



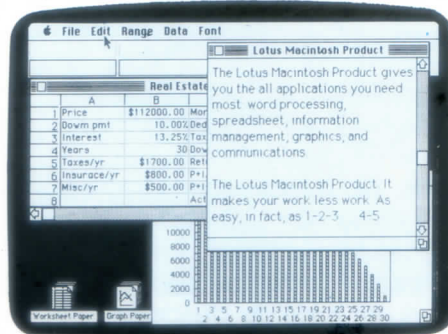
Mac Salários
Gestão de Vencimentos



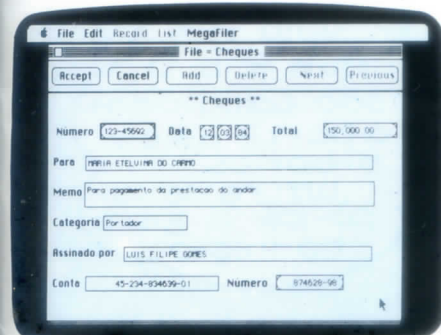
TK! Solver
Cálculo Matemático



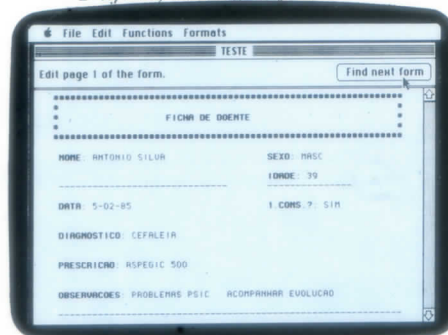
Filevision
Gestão de Ficheiros



JAZZ Gestão Integrada de:
Ficheiros, Folhas de Cálculo,
Gráficos, Comunicações e Texto



MegaFiler
Gestão de Ficheiros



PFS: File/Report
Gestão de Ficheiros e Relatórios



BASIC

K. AZIZ

O curso de BASIC está a chegar ao seu termo e por isso vamos entrar num período de recapitulações que nos parecem importantes, explicando de forma diferente partes fundamentais do nosso curso.

Para abordar a manipulação de informações em ficheiro, utilizaremos uma parte da memória interna do micro computador. Não necessitaremos portanto de suporte exterior de memória caso do leitor/gravador de disquettes. As ideias aqui apresentadas poderão servir no entanto para o iniciar nalgumas técnicas simples de gestão de ficheiros.

Tomemos imediatamente um exemplo concreto:

Quer-se introduzir em memória vários valores numéricos com o objectivo de os analisar sob todas as formas: soma, número de valores negativos, positivos, compreendidos entre 2 determinados valores...

Primeira questão: quantos valores é preciso tratar? Se os contarmos pelos dedos de uma mão, e se soubermos à priori quantos há, pode-se fazer figurar estes valores no programa directamente. Assim:

```
10 PRINT 2+5+7+3
```

uma calculadora faz isto a brincar

Vai calcular e imprimir no display a soma dos quatro números, (sempre os mesmos), e o resultado será sempre...

17

Se previrmos quatro "recipientes", ou seja quatro casas da memória, para guardar os quatro valores, basta dar-lhes um nome:

```
140 PRINT V1+V2+V3+V4
```

e prevê a introdução dos valores, no decurso da utilização do programa:

```
100 INPUT V1
110 INPUT V2
120 INPUT V3
130 INPUT V4
140 PRINT V1+V2+V3+V4
```

quatro variáveis simples: o seu conteúdo pode variar mas cada uma apenas pode incluir um só valor a cada instante

O programa assim obtido interromper-se-á quatro vezes para dar ao utilizador a possibilidade de digitar os valores da sua escolha:

```
RUN
? 2
? 5
? 7
? 3
  17
OK
```

Progredimos: o mesmo programa vai poder tratar qualquer conjunto de quatro informações numéricas, para calcular a sua soma. Se esses números representarem receitas e despesas, poderemos mesmo pedir um total de receitas e um total de despesas:

```
100 INPUT V1
110 INPUT V2
120 INPUT V3
130 INPUT V4
140 PRINT "BALANÇO", V1+V2+V3+V4
150 R=0
160 D=0
170 IF V1<0 THEN R=R+V1 ELSE D=D+V1
180 IF V2<0 THEN R=R+V2 ELSE D=D+V2
190 IF V3<0 THEN R=R+V3 ELSE D=D+V3
200 IF V4<0 THEN R=R+V4 ELSE D=D+V4
210 PRINT "RECEITAS", R
220 PRINT "DESPEASAS", D
```

uma vez que R e D são calculados a partir deles próprios, um reflexo: põ-los a zero no início



cont. BASIC

A utilização deste programa, no caso duma pessoa gastadora, pode dar:

```
RUN
? 2370
? -3650
? 1200
? -2712
BALANÇO -2792
RECEITAS 3570
DESPESAS -6362
OK
```

Mas para tratar cinco valores em lugar dos quatro previstos, será necessário modificar o programa acrescentando uma linha para a introdução do quinto valor, e uma linha para decidir se esse valor faz parte das receitas ou das

despesas, e convém ainda modificar a linha 140 que calcula a soma global. Nada prático, sobretudo se quisermos prever o tratamento, não de cinco, mas de cem valores: seria então preciso um programa de 201 linhas!

Um outro método consiste em reutilizar a mesma casa de memória, a que chamaremos V por exemplo. O tratamento toma então uma forma repetitiva:

```
100 S=0
110 R=0
120 D=0
130 FOR N=1 TO 5
140 INPUT V
150 S=S+V
160 IF V<0 R=R+V ELSE D=D+V
170 NEXT N
180 PRINT "BALANÇO", S
190 PRINT "RECEITAS", R
200 PRINT "DESPESAS", D
```

Handwritten notes:

- cloud: cada valor expulsa um outro
- cloud: é assim necessário tratá-los um a um

```
RUN
? 2370
? -3650
? 1200
? -2712
? 1150
BALANÇO -1642
RECEITAS 4720
DESPESAS -6362
OK
```

Handwritten note:

- cloud: acrescentámos um valor

Mas na linha 30, a introdução de cada novo valor apaga o anterior, uma vez que este é guardado no mesmo lugar. Portanto, não constituímos um ficheiro.

Para o conseguir, vamos utilizar uma variável indexada, ainda chamada quadro, e que não é mais do que um ficheiro armazenado numa zona da memória in-

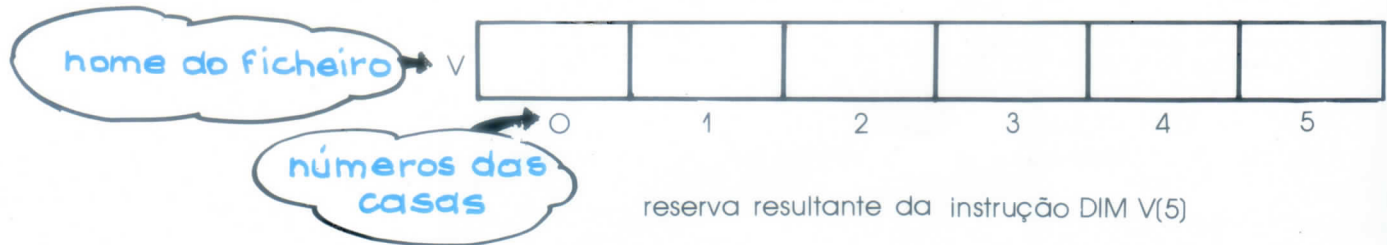
terna do computador.

Em vez de imaginar um nome para cada casa destinada a receber um valor, o BASIC oferece a possibilidade de se associar a um



cont. BASIC

só nome, um número de casa:



O programa passa então a ser:

```
100 DIM V(5)
110 S=0
120 R=0
130 D=0
140 =FOR N= 1 TO 5
150 INPUT V(N)
160 S=S+V (N)
170 IF V(N)>0 THEN R=R+V(N) ELSE D=D+V(N)
180 NEXT N
190 PRINT "BALANÇO", S
200 PRINT "RECEITAS", R
210 PRINT "DESPESAS", D
```

Depois de termos utilizado este programa:

```
RUN
? 2370
? -3650
? 1200
? -2712
? 1150
BALANÇO -1642
RECEITAS 4720
DESPESAS -6362
OK
```

podemos tirar partido duma primeira vantagem desta técnica; a de poder consultar o ficheiro para verificações: chamemos o conteúdo da segunda casa em modo imediato graças a:

a resposta é logo

PRINT V(2)

```
-3650
OK
```



LOGO

CARLOS AMARAL

Introdução

A maioria das linguagens de programação foram criadas numa época em que o custo das memórias era elevado e como consequência a sua capacidade era limitada. Além disso, o acesso aos computadores era restrito às pessoas do "meio informático". Procurava-se então simplificar a linguagem do ponto de vista do computador, mesmo que isso tornasse o trabalho do programador mais difícil. Assim, as linguagens ditas de alto nível aproximavam-se muito da representação física da máquina. Esta ligação entre o "hardware" e o "software" está bem patente nas estruturas de dados utilizadas: apenas se previa a utilização de variáveis de tamanho bem definido (escalares ou matrizes)!

Na década de 60 quando o preço dos computadores começou a descer e um maior número de pessoas os começou a utilizar, foi criada a linguagem de programação BASIC. A ideia dos seus inventores, que é ainda partilhada por muitos, é que, como o BASIC apenas dispõe de algumas instruções de base, é fácil de aprender.

Com a enorme difusão dos microcomputadores a maioria dos quais trabalhando, senão unicamente pelo menos à partida apenas com BASIC, chegou-se a uma situação em que a maior parte das pessoas que sabe programar um microcomputador ou está a aprendê-lo, o faz em BASIC.

No entanto se fizermos uma análise desta linguagem e a compararmos com linguagens posteriores como o Pascal ou o C, encontraremos nela um grande número de deficiências: não há procedimentos, nem verdadeiras funções, não há passagem de argumentos, não há variáveis locais, não é permitida a recursividade, não existem estruturas de dados dinâmicas, estruturas de controlo, etc.

Para ultrapassar estas limitações, quando se pretende fazer

um programa um pouco mais complicado, cai-se inevitavelmente na chamada "técnica de programação em esparquete", já abordada na introdução ao curso de Pascal. (Ver Rev. n.6)

Os leitores mais atentos lembrar-se-ão com certeza de que a única "desvantagem" inerente à utilização duma linguagem como o Pascal é a necessidade dum sistema de discos em grande parte dos casos, para a fácil edição, compilação, ligação e execução dos programas.

A origem do LOGO

A linguagem **LOGO** foi concebida em torno de ideias radicalmente opostas às que deram origem ao BASIC mas, tratando-se duma linguagem interpretada e não compilada, permite uma fácil edição e execução dos programas, facilidade essa que, como veremos, é uma das facetas presentes em todos os aspectos da linguagem.

A sua grande potencialidade e facilidade de emprego, reflectem-se na memória ocupada por um bom editor/interpretador de **LOGO** que é invariavelmente superior ao ocupado pelo BASIC. Este aspecto é actualmente irrelevante, devido à diminuição do preço das memórias e consequente aumento da sua capacidade, o que permite a qualquer dos actuais microcomputadores correr o **LOGO**.

Sobre a direcção de Seymour Papert e em colaboração com o laboratório de inteligência artificial do M.I.T. (Massachusetts Institute of Technology), uma equipa de pesquisa estudou o problema do ensino da informática a pessoas de todas as idades.

Para isso, desenvolveram uma linguagem de programação à qual chamaram **LOGO**. Esta é essencialmente um dialecto do Lisp, uma linguagem muito potente utilizada nas pesquisas no campo da inteligência artificial.

A linguagem **LOGO** foi pensada como a primeira linguagem de programação a ensinar aos interessados. Está provado que a primeira linguagem que se aprende influencia inevitavelmente o modo como se programa, e o modo como se estrutura o desenvolvimento de novos programas, enquanto se programar. Os defensores do **LOGO** afirmam que as bases fornecidas pelo estudo inicial desta linguagem são muito mais apropriadas a um futuro brilhante na programação do que as produzidas pelo BASIC.

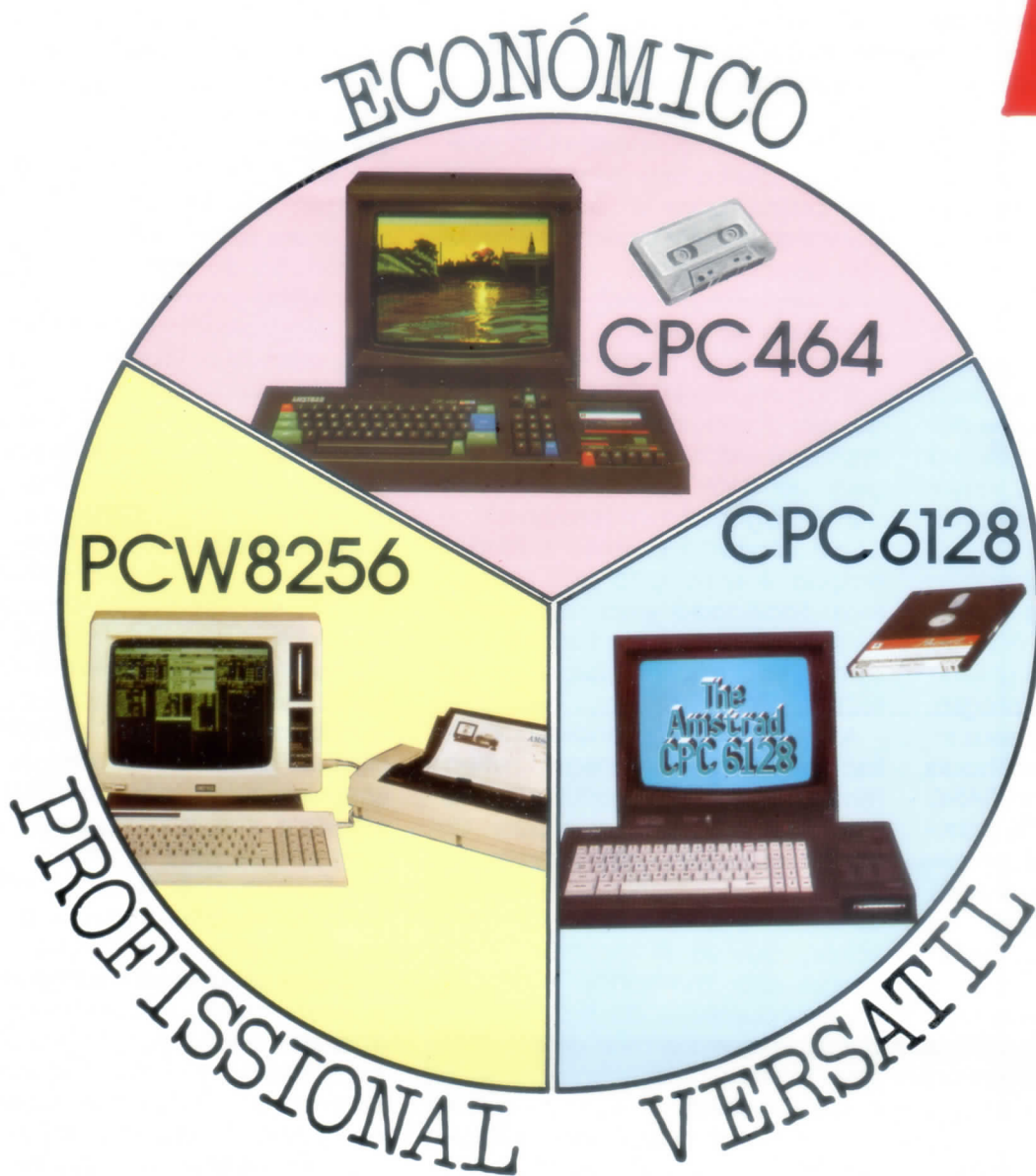
Papert trabalhou durante muitos anos com Jean Piaget, o especialista da psicologia cognitiva. Este observou que as crianças conseguem aprender coisas bastantes complexas — como o falar ou o caminhar — sem um treino formal. O facto desta informalidade estar ausente do ensino tradicional, levou Papert a criar uma linguagem que remediasse esta deficiência. Como ele próprio escreveu no seu livro "Mindstorms":

"Actualmente, num grande número de escolas, a expressão "ensino assistido por computador" significa que o computador está programado para ensinar a criança. Poderia dizer-se que o computador serve para programar a criança. Na minha visão das coisas, a criança programa o computador e, deste modo, adquire o domínio dum dos elementos da tecnologia mais moderna e de mais amplas potencialidades, ao mesmo tempo que estabelece um contacto íntimo com algumas das noções mais profundas da ciência, das matemáticas e da arte de construir modelos intelectuais".

Programação em LOGO

Programar em **LOGO** é fácil: as primitivas, isto é, os procedimentos que constituem a base da linguagem, são potentes e noções reputadamente difíceis como a recursividade ou a passagem de parâmetros, podem ser, com o LO-

A ESCOLHA FOI UNÂNIME
...TEMOS A MAIORIA



A
M
S
T
R
A
D
FAMÍLIA

REPRESENTADO EM EXCLUSIVO
E DISTRIBUIDO POR:

Cominform
COMPANHIA PORTUGUESA DE INFORMÁTICA

LISBOA - AV. DO BRASIL, 147 A-B • TELEF. 80 99 80
PORTO - AV. DA BOAVISTA, 2881, LOJA 3 • TELEF. 68 99 88



cont. LOGO

GO, abordadas por crianças. Ao contrário do BASIC que deixa de lado as noções fundamentais da programação, o LOGO permite aprendê-las em profundidade.

O **LOGO**, à semelhança do FORTH, pode ser visto sob a forma dum dicionário de **primitivas**, que o utilizador enriquece definindo as suas próprias palavras, os seus próprios procedimentos. É o que se chama uma linguagem extensível pois os procedimentos do utilizador utilizam-se da mesma forma que as primitivas.

As novas definições são feitas utilizando a palavra **TO** seguida do nome do procedimento. Entra-se então no modo de edição e introduzem-se as instruções que constituem o procedimento. A instrução **END** significa fim do procedimento e volta a colocar o sistema em modo de execução.

A tartaruga

Para que possamos analisar alguns exemplos de programas em **LOGO** vamos começar por apresentar aquele que é talvez o aspecto mais conhecido da linguagem: a **tartaruga** e os gráficos e a geometria a ela associados.

Esta tartaruga é uma invenção de Seymour Papert. Disse ele:

"Porque não dispor de um pequeno animal algures no écran, uma tartaruga por exemplo, que sabe onde se encontra, para que lado é que está virada, e que é capaz de avançar, de virar e traçar linhas quando se desloca?"

Com este animal é possível desenhar todo o tipo de grafismos com facilidade, sem ter de recorrer às tão pouco manipuláveis coordenadas cartesianas.

Um conjunto de primitivas permite fazer evoluir a tartaruga. As principais são:

FORWARD
BACK
LEFT
RIGHT

para avançar
para recuar
para virar à esquerda
para virar à direita

Com o auxílio destas primitivas é possível traçar já inúmeros grafismos.

Vejamos por exemplo, como podemos definir um procedimento que desenhe um quadrado de lado 100:

```
TO QUADRADO  
REPEAT 4 [FORWARD 100 RIGHT  
90]  
END
```

isto é, repetir 4 vezes o seguinte: avançar 100 unidades em frente e virar 90°, ou seja fazer um ângulo recto. Para o executar basta escrever **QUADRADO**.

Se pretendermos que o mesmo procedimento possa ser utilizado para desenhar quadrados de qualquer dimensão, podemos incluir após o nome do procedimento, nomes de variáveis que serão atribuídas quando este for invocado:

```
TO QUADRADO: LADO  
REPEAT 4 [FORWARD: LADO RIGHT  
90]  
END
```

Agora quando pretendermos desenhar um quadrado escrevemos **QUADRADO 100**, ou **QUADRADO 27**, etc.

Existem ainda outras primitivas como, **PENUP, PENDOWN, PENERASE, PENREVERSE, FENCE, HEADING, POSITION, WINDOW, WRAP, SETPOSITION, SETHEADING, XCOORDINATE E YCOORDINATE**, que permitem um controlo total da tartaruga e a criação de qualquer tipo de desenho.

O Logo e o ensino

O **LOGO** coloca assim à nossa disposição um "**micro-mundo**" geométrico, sobre o qual a criança pode intervir, bem como uma

linguagem de comunicação que lhe permite verbalizar e pôr em evidência os seus próprios modelos de pensamento. Do seu confronto com o "micro-mundo", a criança vai desenvolver as suas estratégias de pensamento, pelos problemas que encontra e é levada a resolver.

O **LOGO** é pois uma tentativa de renovação da educação, definindo novos papéis para o ensino e o aluno. Nesta perspectiva, o aluno busca experiências e o pedagogo, em vez de lhe transmitir um saber, vai procurar facilitar-lhe a sua descoberta propondo à criança "micro-universos" relacionados com esse saber.

Este método de aprendizagem não está de forma alguma limitado às crianças. Existem domínios muito abstractos que podem ser modelados desta forma e propostos a estudantes de todas as idades. Como exemplo, refira-se que dois professores do M.I.T. conceberam um "micro-mundo" regido pelas leis da mecânica relativista, para ensinar física teórica a estudantes universitários.

O LOGO e a Inteligência artificial

Aliás uma ideia bastante vulgarizada é de que o **LOGO** é uma linguagem para crianças que apenas serve para "fazer bonecos"! Não há dúvida de que o **LOGO** tem sido amplamente utilizado nas escolas com crianças em projectos experimentais, os quais têm dado resultados altamente satisfatórios. No entanto, o **LOGO** não é só a tartaruga e a sua geometria.

Além de todas as operações aritméticas e lógicas e funções matemáticas usuais, o controlo de execução, a comunicação com o exterior e com o sistema operativo, existentes em qualquer linguagem, o **LOGO** herdou do Lisp características que lhe conferem enormes potencialidades no campo da inteligência artificial.

Assim, ao contrário das lingua-



cont. LOGO

gens tradicionais, a estrutura de base não é a tabela (ou matriz) mas sim a **lista**.

Uma **lista** é uma sequência ordenada de objectos entre parêntesis rectos:

```
(1 27 CARRO 9)
() (lista vazia)
((CARRO RODA) (BARCO LEME))
```

Os elementos numa lista podem ser palavras, numéricas ou não, ou mesmo outras listas. Em relação às matrizes, as listas têm uma grande vantagem: têm um formato totalmente variável.

A forma como são tratadas as **variáveis** no LOGO é também bastante original. Uma variável pode corresponder a um nome, um número ou uma lista sem necessidade de declaração prévia ou adição dum carácter especial ao nome da variável.

Além disso é permitida a referência indirecta como em Assembly, pois uma variável pode conter o nome dum segunda variável à qual se pode aceder a partir da primeira.

Como **linguagem de manipulação simbólica**, o LOGO está dotado de numerosas primitivas de tratamento de listas. Estas primitivas podem dividir-se em 3 categorias: extracção de informação dum lista, construção de novas listas e testes lógicos. As mais importantes são:

FIRST, que extrai o primeiro elemento dum lista não vazia.

BUTFIRST, que é a operação complementar a FIRST pois dá todos os elementos dum lista, excepto o primeiro.

FPUT, permite inserir um objecto no princípio dum lista.

EQUALP, permite saber se duas listas são iguais (também pode escrever-se "=").

LISTP, indica se um objecto é uma lista.

NUMBERP, indica se um objecto é um número.

EMPTYP, indica se um objecto é uma lista vazia.

Existem muitas outras primitivas para manipulação de listas, mas

estas permitem programar, todas as outras. Vejamos um exemplo:

LAST, que extrai o último elemento dum lista não vazia (:L):

```
TO LAST: L
  IF EMPTY BUTFIRST: L
```

```
(OUTPUT FIRST: L)
OUTPUT LAST BUTFIRST: L
END
```

O que significa que o último elemento dum lista: é ou o primeiro elemento se esta apenas contém um, ou o último elemento da lista que fica quando se retira o primeiro elemento. (O comando **OUTPUT** indica o valor a retornar pelo procedimento).

Uma particularidade que é a chave de um grande número de programas de inteligência artificial e que é quase exclusiva do LOGO e do Lisp é a de se poderem, de uma certa forma, **auto-programar**. Com efeito, os procedimentos em LOGO são representados sob a forma de listas e podem ser manipulados como dados.

Assim, por manipulação de listas, são criados procedimentos, que podem em seguida ser executados com o auxílio do comando **RUN**. É pois possível a um programa criar procedimentos e executá-los.

Graças a esta característica, podem ser criadas estruturas de controlo adaptadas às necessidades. Vejamos por exemplo, como poderíamos definir as estruturas **REPEAT** e **WHILE**, aplicadas à lista de instruções contida em :L:

```
TO REPEAT :N :L
  IF :N=0 (STOP)
  RUN :L
  REPEAT :N-1 :L
END
```

```
TO WHILE :COND :L
  IF NOT RUN :COND (STOP)
  RUN :L
  WHILE :COND :L
END
```

Todas as possibilidades são permitidas; apenas é necessário um pouco de imaginação. O LOGO é mesmo visto por alguns como uma espécie de "micro-mundo" dos informáticos!

A representação de conhecimentos

Uma das tarefas sobre as quais se debruça a inteligência artificial é a análise de fenómenos ou situações do nosso mundo. Trata-se então de representar as relações qualitativas existentes entre pessoas, objectos e outras entidades do nosso universo quotidiano.

Um método simples de representar estes objectos e as suas relações, é através da utilização de **listas de propriedades** ("property list" ou "P-list") associadas às palavras do LOGO.

À semelhança do que acontece no PROLOG, estas listas de propriedades permitem associar a uma palavra, uma qualidade e o seu valor.

Por exemplo, se quisermos dizer que o Luís é amigo do Paulo e do Zé, basta colocar a lista (Paulo Zé) na lista de propriedades do Luís, sob o título "AMIGO". Em LOGO, teremos:

```
PPROP "Luís" AMIGO
(Paulo Zé)
```

Para reter esta propriedade podemos fazer

```
GPROP "Luís" AMIGO
que produzirá
(Paulo Zé)
```

É ainda possível criar procedimentos que definem que uma dada relação é **simétrica**; isto é, se o Luís é amigo do Paulo, concerteza o Paulo é amigo do Luís; ou **inversa**, isto é, se o Zé possui um carro, o carro é do Zé (ou possuído pelo Zé), etc. Podem ainda introduzir-se as **relações transitivas**, com precaução pois a sua programação conduz frequentemente a "ciclos infinitos".

Depois de introduzidos todos os objectos e respectivas relações, podem ser criados procedimentos

cont. na pág. 58



1ª enic

O público acorreu em elevado número durante os quatro dias da 1ª ENIC.



Nem sempre o número é o mais importante, daí que nalguns stands se lamentasse o facto de não ter sido reservado um dos dias só para profissionais. Era no entanto de optimismo o sentimento geral dos expositores quanto aos dividendos a tirar da Feira, e da parte dos visitantes também cremos que não foram goradas as expectativas.

Sabemos como o público em geral, aprecia estas ocasiões de "convívio" com equipamentos a que de outro modo dificilmente tem acesso, com o aliciante extra das novidades que se possam encontrar ou quando muito que se possam ouvir anunciar para breve.

Actualização é o termo que melhor exprime o que o visitante espera destas realizações, por isso pensamos que quem esteve na ENIC não se sentiu defraudado.

Num excelente espaço, uma organização atenta, e a plena adesão dos expositores (houve que re-

jeitar a participação de empresas, por falta de espaço) e do público, assim decorreu a 1ª Exposição Nacional de Informática, Comunicação, Burótica e Microfilme.

Paralelamente à exposição decorreram nos auditórios do Forum palestras dedicadas à informática e sessões cinematográficas acerca da produção de imagens a 3 dimensões por computador.

Estas sessões cinematográficas foram promovidas pela API e permitiram ficar a conhecer o que de melhor se faz nesta área em todo o mundo. Imagens de extrema perfeição!

De parabéns está a **API** por nos proporcionar tão bons momentos "informáticos".

A Visita

É agradável reparar que às vezes, Portugal não está tão longe do resto do mundo como se diz por aí. Foi uma surpresa encontrar o **ATARI 520ST** na ENIC, cujo lançamento só está previsto para princípios de 86.

Para os mais curiosos, o ATARI 520ST é um modelo baseado no microprocessador 68000 da MOTOROLA e a sua configuração básica é: 198 K ROM + 512 K RAM + 360 K em disquete formatada de 3,5". No entanto o modelo de pré-

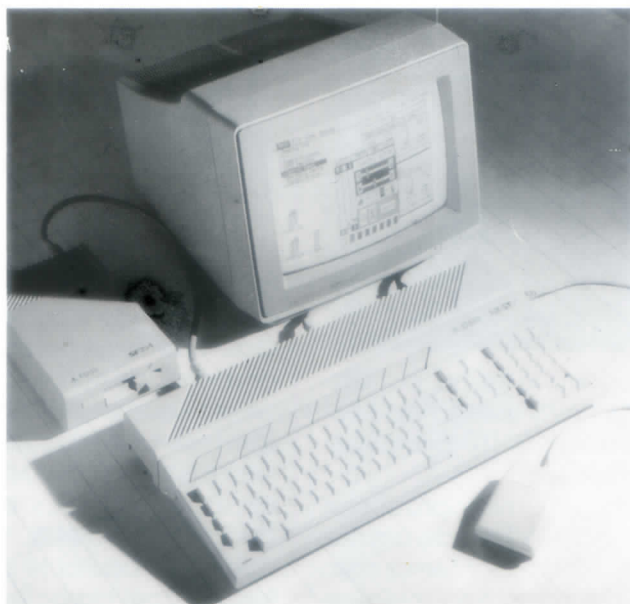
-lançamento que estava na ENIC tinha 16 K ROM + 512 K RAM, 250 K dos quais ocupados pelo sistema operativo, o que nos deixa "apenas(!)" cerca de 250 KBYTE para trabalhar.

O sistema operativo é o TOS-TRAMIÉL OPERATING SYSTEM. Nele está incluído o GEM-GRAPHICS ENVIRONMENT MANAGER, sistema de exploração gráfica criado pela DIGITAL RESEARCH. O GEM foi desenvolvido para o IBM PC e compatíveis, posteriormente adaptado ao APRICOT (havia aliás um APRICOT F1 e funcionar com o GEM na ENIC) e agora no ATARI 520ST.

O GEM tem um funcionamento parecido com o do Macintosh, mas internamente nada tem a ver com ele. É um sistema de Software puro, tão geral que até pode funcionar em máquinas baseadas no 8086 da INTEL.

O sistema da Macintosh está intimamente relacionado com o seu Hardware sendo-lhe portanto, totalmente particular.

O Software agora disponível limita-se a compiladores, mas já estão na forja muitos programas que, a estarem prontos na data de lançamento, tornarão o ATARI 520ST um verdadeiro competidor.

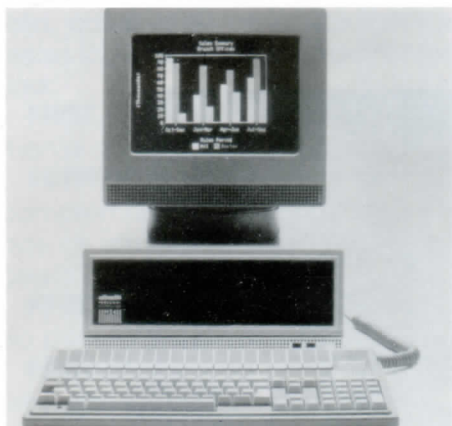




A **SPERRY** apresentou o seu novo **PC IT**, um microcomputador que corre o sistema operativo XENIX e portanto com possibilidade de ligar vários terminais.

Tinha também em exposição um sistema da série Sperry 5000, baseado no sistema operativo UNIX. Finalmente, ao sistema de escritório **SPERRYLINK**, vem agora juntar-se um leitor de documentos, que regista automaticamente documentos escritos à máquina, eliminando a necessidade de o fazer manualmente.

Dimensões reduzidas e facilidade de utilização são os "argumentos" do Leitor de Documentos Sperrylink.



Em frente estava o stand da **OLIVETTI** com a nova versão do M24 denominada M24SP (SP para Special Performance).

Dotado de um microprocessador Intel 8086 a 10Mhz e de um canal de comunicação interna com uma maior capacidade, oferece

uma velocidade de cálculo mais elevada, com tempos de acesso extremamente curtos.

Presentes estiveram também os micros da **BULL**.

A CeBIT sempre atenciosa, mostrou com orgulho os Micral 90/30 e 90/50. O primeiro é compatível IBM enquanto o 90 vai muito mais longe, tendo capacidade inata de multiposto.



Dr. Hermano Ferreira, da **BULL**

A compatibilidade com os programas para o IBM PC em MS/DOS é total, tendo sido feitas experiências com programas e diskettes preparadas na IBM.

Essa compatibilidade tem origem num sistema operativo muito interessante, utilizado até agora em apenas outros dois computadores: **PROLOGUE**. É um sistema operativo bastante simples e evoluído e com características de tra-

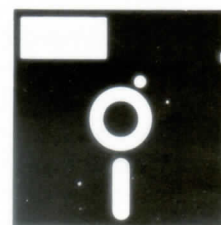
balho multiposto até 4 computadores: o Micral 90 mais três Micral 30. Um ponto a salientar é a base de dados relacional **DIALOGUE**, facilmente configurável e muito potente.

A ter em atenção.

Apenas uma nuvem no horizonte: a continuidade destes produtos que, apesar de bons, têm pouca implantação mundial.

No stand da **NPN** um produto houve que passou despercebido e que interessa fazer referência: estamos-nos a referir ao **HYPERDRIVE**, um disco rígido para o Macintosh, representado em Portugal pela Interlog

O disco rígido é montado dentro do Macintosh. O seu controlador tem acesso directo ao coração daquele: o microprocessador 68000. Como passa por cima de qualquer tipo de Interface de comunicação tem muito mais eficiência, aumentando em 15 vezes a velocidade do Macintosh.



software

A **NCR PORTUGAL** apresentou na ENIC novos sistemas informáticos nas linhas de computadores pessoais, sistemas UNIX tm, Sistemas Industriais e Caixas Automatizadas (ATM).

Destaque para os novos computadores pessoais que incluem o **PC 4 I**, um 16 bits compatível PC/XT, o **PC6**, também compatível mas oferecendo elevadas performances e o **PC8**, o primeiro PC-AT compatível anunciado no mercado português.



A nova Caixa Automatica (**ATM 5070**) vem reforçar a imagem NCR no dominio da fiabilidade dos seus sistemas, apresentando características únicas na comunicação com o cliente (ecran de gráficos, privacidade, módulo de voz, etc.) e permitindo ciclos de reabastecimento mais alongados.

Foi oficialmente anunciado na Feira a representação da **COMPUTER VISION** pela **PORTU**.

As razões que justificam esta decisão foram apontadas por um director da Computer Vision numa pequena declaração: dinamismo, juventude, profissionalismo.

Neste momento a Portu possui 3 programas de CAD-Computer Aided Desing bastante completos e muito profissionais, da Computer Vision, claro. Um de projecto de circuitos impressos, um de desenho simples a três dimensões e outro de projecto de interiores de arquitectura.

A Portu fornece ainda suporte fisico para os programas, baseado no IBM PC especialmente configurado e equipado de periféricos para o efeito.



Eng. Coucello, da Olivetti e o Dr. Raul Junqueiro Luis Oliveira, da "Cérebro".



Dr. Raul Junqueiro e o Eng. Miguel Prieto, da Burroughs.

TIMEX COMUNICADO

Têm sido recentemente publicados na Imprensa anúncios de "Campanha de Material da T.C.", publicitando a venda dos nossos produtos TC2048, TC2068 e Spectrum TIMEX, a preços que, a serem satisfeitas as exigências legais e as necessidades comerciais, não podem ser praticados.

Para esclarecimento do público em geral e dos seus Clientes em particular, a TMX PORTUGAL, LTD informa que as referidas actuações são desenvolvidas pela TRIUDUS, Lda., empresa à qual foram há muito cortados os fornecimentos directos daqueles produtos, por falta de pagamento.

A TMX PORTUGAL LTD iniciou já e vai continuar, as diligências necessárias a pôr cobro e punir tais actuações, recorrendo nomeadamente às instâncias judiciais competentes.

O Representante Legal

PERSONAL COMPUTER

OLIVETTI:

A PERFORMANCE

DA FORMULA 1.

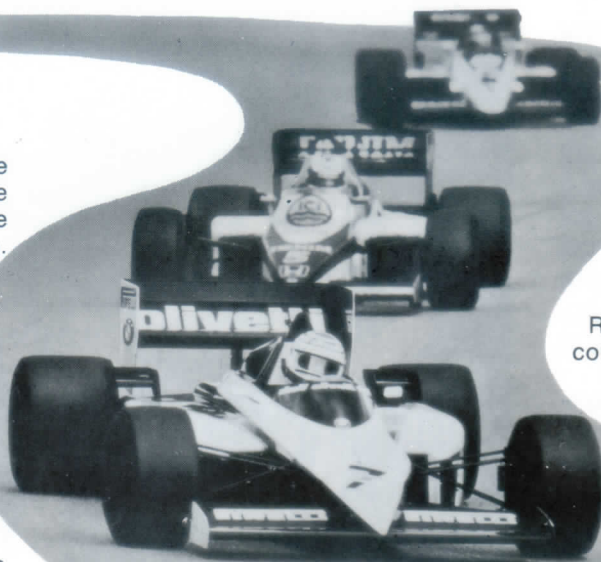


Quem vê as corridas na Televisão pode pensar que o "Personal Computer" que trabalha na Fórmula 1 tem apenas que calcular os tempos. Porém, não é assim. Porque a Fórmula 1 é, entre todos os clientes, o mais difícil: procura sempre novas soluções sem nunca se contentar.

Hoje ao "Personal Computer" Olivetti pede-se a determinação do ponto de máxima velocidade de cada circuito. E a passagem de dados, em tempo real, da cronometragem para todas as equipas. E comparações, valorizações, hipóteses sobre cada uma das variáveis. E no fim de um dia de treinos ou de competição o somatório de todas as informações para o perfeito funcionamento das viaturas.

Assim, no fundo, por detrás de cada bom resultado na Fórmula 1 está um pouco do nosso trabalho e em cada "Personal Computer" Olivetti está todo o aperfeiçoamento requerido pela Fórmula 1. Quantas vezes o seu escritório se transforma numa pista de Fórmula 1 e quantas vezes no seu trabalho se pedem "performances" de Fórmula 1? Leve consigo um "Personal Computer" Olivetti e poderá constatar imediatamente como pode tratar mesmo os problemas mais específicos.

Com uma velocidade de cálculo excepcional. Com uma biblioteca de programas praticamente ilimitada, graças à compatibilidade com os "standard". Com uma gama de soluções "hardware" correspon-



dentes às mais diversas necessidades.

Com uma resolução gráfica de nível superior com uma legibilidade do ecrã que não tem comparação.

Eis porque os "Personal Computer" Olivetti foram escolhidos pela Fórmula 1 e estão a conhecer na Europa e também nos mercados mais competitivos, como os Estados Unidos, um sucesso que não pára de crescer.

Enfim, um "Personal Computer" Olivetti merece mesmo ser experimentado.

Dirija-se aos Agentes "Dealers" Olivetti que formam uma das redes de distribuição, consultadoria e assistência

mais vastas de Portugal: especialistas altamente profissionais, sempre prontos a explicar, aconselhar, resolver, propôr.

Ou então envie-nos este "coupon". Receberá mais informações e poderá constatar que num "Personal Computer" Olivetti está toda a tecnologia de quem está habituado às competições mais duras e sabe fornecer "performances" de

Fórmula 1.
Todos os dias consigo no seu trabalho.

Para mais informações enviar à OLIVETTI ✂

Nome:

Morada:

Localidade:

Telef:

OLIVETTI — Divisão Personal Computer
Largo S. Sebastião da Pedreira, 32-B — 1000 LISBOA — Tel: 534137

olivetti



Dr. Raul Junqueiro e o Eng. Zagalo Lima, da Xerox.



Dr.ª Célia Metraço, Dr. Raul Junqueiro e o representante da API, Dr. Pereira da Costa.

Reportagem fotográfica de:
Estudios Homem Cardoso

A **RANK XEROX**, sempre eficiente e profissional, nada divulgou sobre os futuros lançamentos que grande curiosidade começam já a despertar. De concreto apenas a intenção de 55% dos lucros em 86, a nível mundial, ser decorrente do sector de informática o que representa sem duvida, e se assim lhe pudermos chamar, uma autêntica viragem na política da empresa. Vamos estar pois atentos.

A **TELEMÁTICA** apresentou na ENIC um conjunto de programas de gestão integrada para as empresas. Destinam-se ao RAINBOW da Digital e são os seguintes:

- contabilidade geral
- gestão de stocks
- gestão de terceiros
- gestão de pessoal e salários
- facturação



No nosso stand

Foram bastante movimentados aqueles quatro dias da ENIC. É sem duvida extremamente benéfico o convívio e o contacto pessoal com os nossos leitores, e esta foi mais uma boa ocasião para que isso acontecesse. No decorrer da Feira muitos leitores nos visitaram, transmitindo-nos simpáticas palavras de apoio e incentivo, apresentando-nos sugestões, pedindo esclarecimentos.

A perfeita identificação entre a revista e o seu leitor é um objectivo só atingido com o decorrer do tempo, por isso são importantes os contactos directos com aqueles que nos lêem.

A enorme procura dos números atrasados e as assinaturas que se fizeram, à falta de outros indicadores, era suficiente prova do agrado registado pelo nosso trabalho.

Em especial para os leitores da região do Porto relembramos que vamos estar presentes na INFORPOR 85.

Esperamos a sua visita!

INFORPOR 85
INFORPOR 85
INFORPOR 85
INFORPOR 85
INFORPOR 85
INFORPOR 85
INFORPOR 85

2ª EXPOSIÇÃO PORTUGUESA
DE INFORMÁTICA
1ª EXPOSIÇÃO DE TECNOLOGIAS
DA INFORMAÇÃO
5-8 DEZEMBRO/85
PALÁCIO DE CRISTAL / PORTO



ASSOCIAÇÃO PORTUGUESA DE INFORMÁTICA **CERTAME** feiras exposições e congressos, lda.

Para mais informações: R. do Arco do Carvalho, N.º 1-2.º D-1000 Lisboa
Telf. 65 75 20 / 65 75 24 / 65 75 88 - Telex 64277 CERTAM P
Porto - Telf. 69 84 94 / 69 53 85



BIORITMO

J. MATON

O bioritmo é uma teoria de origem americana que afirma que cada um de nós passa por períodos "altos" e "baixos", no decorrer dos quais se está mais ou menos em forma.

Os percursos desta teoria defendem que esses períodos se organizam em ciclos regulares tendo origem na nascença do indivíduo.

É desta forma possível, se se calcular a evolução desses ciclos até a uma determinada data, determinar com precisão o nível em que se encontra o indivíduo nesse momento.

Os três ciclos do bioritmo são:

- Físico (FIS): 23 dias
- Emocional (EMO): 28 dias
- Intelectual (INT): 33 dias

Cada um destes ciclos é representado por uma curva sinusoidal cujo período é igual à duração do ciclo e a origem se situa no ponto zero (dia do nascimento).

O programa proposto permite calcular os diversos pontos das três sinusoides (ordenadas), o número de dias escolhidos está disposto em abcissa.

Note-se que este programa permite ainda o cálculo do dia da semana assim como o número de dias compreendido entre duas datas determinadas.

Por outro lado é possível, sem dificuldade, modificar o programa para tirar uma listagem na impressora.

LOGIN

“SOFTWARE-HOUSE” UMA LOJA NOVA COM IDEIAS NOVAS

— Cursos de programação BASIC E ASSEMBLER

— SOFTWARE — jogos, programas profissionais, utilitários e de ensino, em cassette ou diskette
para : Spectrum, TC 2048, TC 2068 e Spectrum +
Commodore
QL
IBM PC
EPSON

e o nosso primeiro exclusivo, um “package” de informatização de firmas de Seguros

— Cheque SOFTWARE DE NATAL, a solução para as suas prendas

— Brevemente o lançamento do 2068 CARTRIDGE CLUB

— Software por encomenda, e uma VASTA GAMA de programas originais para o sistema FDD

— SOS técnico por telefone — coloque-nos as dúvidas, nós apresentamos-lhe a solução

— Biblioteca Software

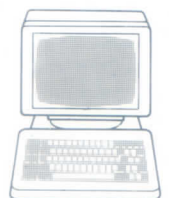
SOMOS UMA NOVA EXPRESSÃO NO ATENDIMENTO CONTACTE-NOS

CENTRO COMERCIAL STROMP (Junto ao Estádio de Alvalade)

LOJA 8

Telef. 7584445

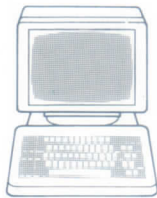
10 PRINT "BIORRITMO"	TECLA: "	
20 REM PROGRAMA DE J. MATON	60 PRINT "1- PARA TER O SEU BIORRITMO"	
PARA EPSON HX 20	70 PRINT "2- PARA SABER O DIA DA SEMANA DE UMA DADA DATA"	
30 REM COPYRIGHT O AUTOR E SOFTWARE	80 PRINT "3- PARA SABER O NUMERO DE DIAS ENTRE 2 DATAS"	
40 PRINT "MENU"		
50 PRINT "CARREGUE NUMA		



TANDY
DRAGON 32



cont. BIORITMO



TANDY
DRAGON 32

```

90 PRINT " 4- FIM"
100 INPUT "QUAL A SUA ESCOLHA ?";A2
110 IF A2<1 OR A3>4 THEN PRINT "ERRO":GOTO 40
120 PRINT
130 ON A2 GOTO 150,520,680,890
140 END
150 CLS
160 INPUT "INTRODUZA O SEU NOME: ";X#
170 PRINT "INTRODUZA A SUA DATA DE NASCIMENTO"
180 GOSUB 820
190 CLS
200 IF M<=2 THEN X=W
210 IF X>2 THEN X=W
220 PRINT
230 J=1
240 PRINT "ENTRE O MES E O ANO QUE QUER O BIORRITMO"
250 GOSUB 840
260 IF M=1 OR M=3 OR M=5 OR M=7 OR M=8 OR M=10 OR M=12 THEN N=31
270 IF M=4 OR M=6 OR M=9 OR M=11 THEN N=30
280 IF M=2 THEN N=28+(A/4-INT(A/4))
290 IF M<=2 THEN Y=W
300 IF M>2 THEN Y=Z
310 F=Y-X
320 CLS
340 PRINT
350 PRINT "*****"
360 PRINT "*** BIORRITMO ***"
370 PRINT "*****"
375 PRINT
380 PRINT "REALIZADO PARA";X#
390 PRINT
395 PRINT "PARA O MES DE :";M;"/";A
396 PRINT
400 PRINT "D";TAB(5);"FIS";TAB(12);"EMO";TAB(19);"INT";TAB(25);"MEDIA"
410 PRINT
420 REM CALCULO DO BIORRITMO DO MES
430 FOR K=0 TO N-1
440 P=INT(100*SIN(2*3.1416*(K+F)/23))/100
450 E=INT(100*SIN(2*3.1416*(K+F)/28))/100
460 I=INT(100*SIN(2*3.1416*(K+F)/33))/100
470 M=INT(100*(P+E+I)/3)/100
480 PRINT K+1;TAB(4);P;TAB(11);E;TAB(18);I;TAB(24);M
490 NEXT K
500 GOTO 40
510 STOP
520 CLS
530 GOSUB 820
540 IF M<=2 THEN H=W
550 IF M>2 THEN H=Z
560 REM DETERMINACAO DO DIA DA SEMANA
570 D=H+(INT(1-H/7)*7)
580 IF D=7 THEN K#="SABADO"
590 IF D=1 THEN K#="DOMINGO"
600 IF D=2 THEN K#="SEGUNDA"
610 IF D=3 THEN K#="TERCA"
620 IF D=4 THEN K#="QUARTA"
630 IF D=5 THEN K#="QUINTA"
640 IF D=6 THEN K#="SEXTA"
650 PRINT "DIA";J;"/";M;"E UM(A)";K#
660 GOTO 40
670 STOP
680 CLS
690 PRINT "INTRODUZA A PRIMEIRA DATA"
700 GOSUB 820
710 IF M<=2 THEN X=W
720 IF M>2 THEN X=Z
730 PRINT
740 PRINT "INTRODUZA A SEGUNDA DATA"
750 GOSUB 820
760 IF M<=2 THEN Y=W
770 IF M>2 THEN Y=Z
780 F=ABS(Y-X)
790 PRINT "O NUMERO DE DIAS ENTRE DUAS DATAS E: ";F
800 GOTO 40
810 STOP
820 REM CALCULO DO FACTOR
830 INPUT "DIA: ";J
840 INPUT "MES: ";M
850 INPUT "ANO: ";A
860 W=(365*A)+J+(31*(M-1))+INT((A-1)/4)-INT((3*((A-1)/100)+1)/4)
870 Z=(365*A)+J+(31*(M-1))-INT((0.4*M)+2.3)+INT(A/4)-INT((3*(A/100)+1)/4)
880 RETURN
890 END

```




Novidades Editoriais



O Feiticeiro da Montanha de Fogo

Editora VERBO

Conjunto livro e cassette para o ZX Spectrum.

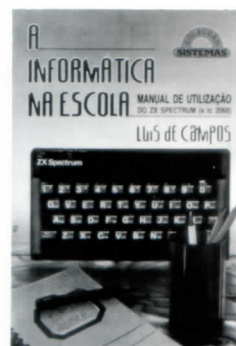
Um livro original, em que você e os dados decidem o desenrolar da acção, uma história diferente em que se joga o seu destino. É uma arriscada expedição em busca do tesouro do feiticeiro. Pode não sobreviver à primeira exploração. Mas com alguma experiência, habilidade e sorte, verá que após cada nova tentativa se encontrará mais próximo do seu objectivo.



O Computador Pessoal IBM e Sistemas Compatíveis

Editorial PRESENÇA

Este é um manual destinado a familiarizar o utilizador do IBM PC. Os autores (Carlos Reis e João Capaz) expõem de forma clara e completa tudo o que há de mais relevante a indicar sobre o hardware, o software, a estrutura e as funções dos sistemas operativos, e a programação Basic. De realçar que todas as informações técnicas se aplicam igualmente a toda a gama de microcomputadores compatíveis com o sistema IBM.



**— A Informática na Escola —
Manual de Utilização do
ZX Spectrum (e tc 2068)**

Editorial PRESENÇA

Um completíssimo manual orientado para diferentes aplicações da informática na escola, abrangendo desde aspectos de organização até aos de carácter pedagógico.

Eis aqui uma edição muito rica e muito útil, não só pelo conteúdo concreto, mas também pelo mérito que tem, de lançar pistas para a exploração das ilimitadas potencialidades da informática, ao serviço do ensino e da formação de jovens.



Microcomputadores

Editora VERBO

Este livro é uma introdução clara ao estudo dos microcomputadores, que acentua particularmente a forma como trabalham. Explica o sistema binário e dá uma ideia sobre os periféricos à disposição dos utilizadores dos microcomputadores. Alguns dos capítulos tratam da pastilha de silício, do software e da programação.



**Inteligência Artificial
no seu Spectrum e Spectrum +**

Editorial PRESENÇA

Desde os programas que aprendem e raciocinam, áqueles que falam consigo, executam ordens e lhe dão bons conselhos, este livro vai guiá-lo pelo fascinante mundo da Inteligência Artificial.

Prepare-se portanto para iniciar esta emocionante viagem entre o mundo das realidades científicas e os horizontes a perder de vista da ficção!

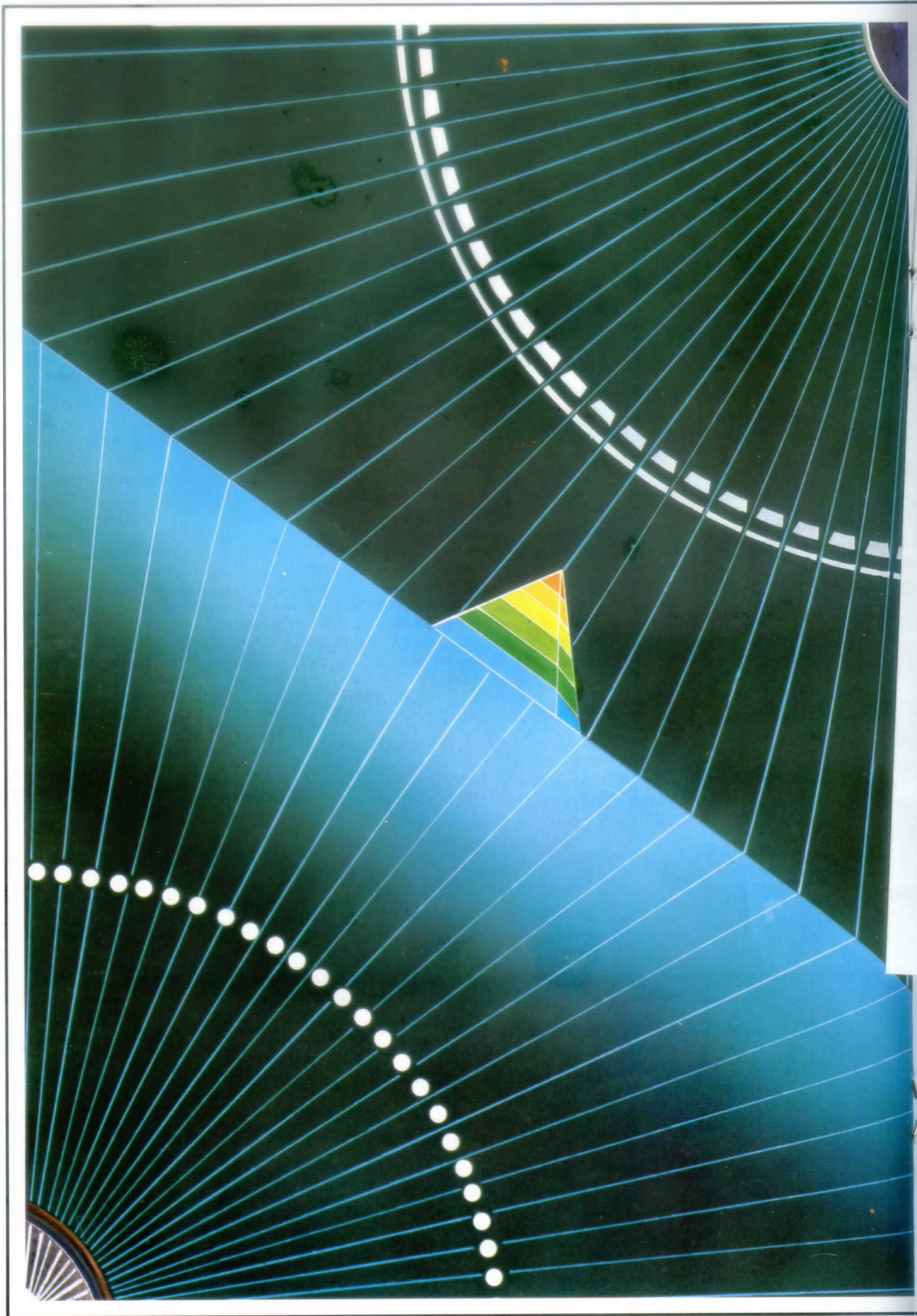


**Os 20 Melhores Programas
para o ZX Spectrum**

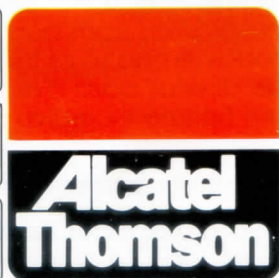
Editora VERBO

Escrever bons programas é uma arte que se pode aprender (os leitores da SOFTWARE são disso testemunha), e o propósito desta obra é ensiná-lo por meio de exemplos escolhidos.

Os vários programas são fonte de referência de técnicas muito difundidas no mundo da informática: alguns deles ilustram por exemplo, o uso de registo de comprimento fixo ou de comprimento variável, a sua ordenação, a pesquisa binária; outros apresentam técnicas especialmente adaptadas ao Spectrum, atribuindo-se especial importância ao uso adequada da imagem.



NO MUNDO DAS TELECOMUNICAÇÕES DIGITAIS,
A TECNOLOGIA DE PONTA TEM UM NOME



- O grande grupo europeu ALCATEL THOMSON nasceu da convergência dos dois leaders no domínio das telecomunicações — CIT-ALCATEL e THOMSON CSF.
- A ALCATEL THOMSON concebe, realiza e faz funcionar todos os tipos de redes de telecomunicações públicas ou privadas.
- Na comutação telefónica digital, a ALCATEL THOMSON é o primeiro construtor mundial: 18 milhões de linhas encomendadas por 45 países, das quais 6 milhões já em serviço.
- A ALCATEL THOMSON está na primeira linha dos grandes construtores mundiais em transmissões por cabos terrestres, fibras ópticas e feixes hertzianos, em ligações intercontinentais submarinas e espaciais, em redes de distribuição de dados e de videocomunicação.
- Das redes "Chaves na mão", à cooperação industrial, ALCATEL THOMSON é uma experiência, um potencial de investigação e desenvolvimento e um know-how que oferece as melhores garantias de um investimento duradouro.

com Alcatel Thomson
O AMANHÃ É HOJE
na rede telefónica portuguesa

ALCATEL THOMSON INTERNACIONAL
33, Rue Emeriau
75725 PARIS CEDEX
II Tel: 571.10.10
Telex: ALCIT 250927 F

CGE ALSTHOM AGMEL, LDA.
Avenida Elias Garcia, 123-5.º
1000 LISBOA
Tel. 73 00 31
Telex: 12463 CGE ATP

Unitros

organizadores

NPN e SOFTWARE

concorrente

Você

meta

Gestão Industrial Optimizada

pole position

Programação



Grande Prémio que vos propomos pretende ser um desafio à capacidade, imaginação e criatividade de todos os programadores portugueses, procurando-se obter uma aplicação prática. Que tipo de Aplicação?

O projecto deve ter aplicação concreta no meio industrial português.

Portugal é um país sui-generis quanto ao seu tecido industrial. Soluções óptimas noutros países geralmente não se aplicam à maioria das indústrias portuguesas.

Assim **«GESTÃO OPTIMIZADA INDUSTRIAL»** pretende ser soluções de portugueses para Portugal.

Para só falarmos na razão mais próxima, a entrada de Portugal na comunidade europeia, é urgente modernizar a nossa indústria. É certo que vão chegar do estrangeiro outros meios, outras soluções, por isso este Grande Prémio pretende ser uma alternativa, que acreditamos vai ser a melhor!

Vai ser como dissêmos «soluções de portugueses para portugueses» o que estamos certos é uma garantia.

Como já foi referido por diversos governantes um dos objectivos nacionais é criar e incentivar uma indústria de software (é impossível fazê-lo a nível de hardware), como única forma de acompanhar a 3.ª Revolução Industrial.

Mesmo que indirectamente esse é também um

campo que será beneficiado com esta iniciativa NPN/revista SOFTWARE. De tal forma que todos os programas com qualidade para serem comercializados (e não só os três premiados), o serão com 25% de direitos para o autor, o que constitui certamente um óptimo incentivo suplementar.

CONDIÇÕES DE PARTICIPAÇÃO

Para que todos tenham à partida as mesmas hipóteses, não restringimos a participação a nenhum equipamento, a nenhuma linguagem, tão somente se exige que o programa funcione com um máximo de 256 K e com um suporte de disco 10 MB, em sistemas operativos UNIX, CPM 86 ou MS/DOS.

Os programas a apresentar podem ser só em módulos tais como:

- Operações de produção
- Seguimento da produção (tempos e materiais)
- Planeamento
- Aproveitamento de áreas
- Outros

Ou de forma integrada.

Três factores importa tomar em conta em gestão industrial optimizada:

- matéria prima ou componentes do produto final
- tempo e custos de máquina
- tempo e custos de mão de obra

NÃO SE PRETENDE programas de gestão de stocks, contabilidade ou salários, que por nós são considerados programas para o sector terciário e portanto de serviços, **A MENOS QUE** estejam englobados de forma integrada na gestão industrial.

Será provavelmente útil para os concorrentes, a fim de se inteirarem das necessidades ou do que se exige numa gestão industrial, que se informem junto de profissionais ligados ao sector.

Grande Prêmio

software



Os trabalhos apresentados podem ser individuais ou de grupo, podem dizer respeito a um ramo da indústria em especial ou limitar-se a funções genéricas. Tudo depende de si, estimado leitor!

NOTA:

Uma vez que podem surgir dúvidas sobre a matéria que cabe num trabalho deste gênero, ou para evitar qualquer outro tipo de problemas que o façam hesitar e portanto o impeçam de rapidamente começar a trabalhar no programa, facultamos-lhe a possibilidade de nos enviar para aprovação, um resumo ou uma descrição daquilo que se propõe fazer, evitando assim erros de interpretação sobre aquilo que lhe é pedido.

REGULAMENTO

- Art.º 1 – O Grande Prêmio é aberto a todos e não está ligado a nenhuma obrigação de compra. É interdito aos colaboradores da revista SOFTWARE e da firma NPN.
- Art.º 2 – O Grande Prêmio tem início a 5 de Julho e a data de entrega dos trabalhos termina a 5 de Janeiro. Nenhum trabalho que chegue depois desta data será considerado, sendo a data do carimbo dos correios que conta.
- Art.º 3 – Os resultados serão divulgados no número de Março da SOFTWARE.
- Art.º 4 – As decisões do Juri não têm recurso.
- Art.º 5 – Os Prêmios não podem ser trocados por dinheiro.
- Art.º 6 – Os concorrentes autorizam a eventual publicação dos seus trabalhos, desde que devidamente identificados.
- Art.º 7 – Os vencedores autorizam a eventual citação dos seus nomes e moradas em qualquer documentação de promoção.
- Art.º 8 – Os organizadores não serão tidos por responsáveis se, por motivos de força maior ou por qualquer outra razão independente da sua vontade, o concurso deva ser adiado, anulado ou modificado. Igualmente não são responsáveis por percas, avarias ou demoras verificadas nos serviços dos Correios.
- Art.º 9 – A participação no Grande Prêmio implica ter concordado com este Regulamento.

prêmios/prêmios/prêmios/prêmios/prêmios/prêmios/prêmios/

1.º PRÊMIO

Um computador SIRIUS
no valor de **669.000\$00**

2.º PRÊMIO

Uma impressora EPSON
no valor de **95.000\$00**

3.º PRÊMIO

Uma viagem a LONDRES
no valor de **40.000\$00**

Os três programas premiados e todos os considerados pela NPN com possibilidades de serem comercializados, sê-lo-ão, com 25% de ROYALTIES sobre as vendas efectuadas.

prêmios/prêmios/prêmios/prêmios/prêmios/prêmios/prêmios/





LINGUAGEM MÁQUINA

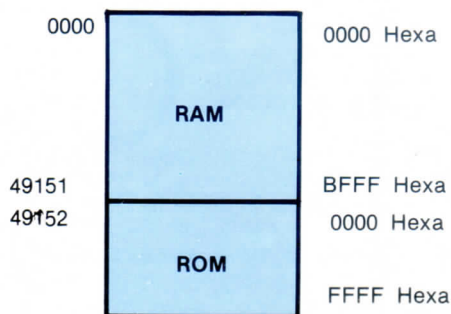
J. AUQUIER

Esta lição é consagrada ao estudo da configuração da memória de um Apple II.

1) Configuração da memória

Esta memória compreende 48 K bytes de RAM e 16 K bytes de ROM, ou seja $48 \times 1024 = 49152$ bytes de RAM e $16 \times 1024 = 16384$ bytes de ROM.

O número total de bytes é portanto de $16384 + 49152 = 65536$ bytes (64 K bytes).



A ROM (Read Only Memory)

A ROM compreende um conjunto de sub-rotinas "Máquina" que constitui o BASIC OPERATING SYSTEM.

Logo que uma instrução "BASIC" deva ser executada, o sistema escolhe os sub-programas que serão postos em acção.

Vamos estudar nos próximos números, as principais rotinas contidas na "ROM".

Para poder executar estes sub-programas, são necessárias algumas posições da memória RAM. Vamos então ver quais as zonas de RAM utilizadas pelo sistema operativo e quais as zonas livres para serem utilizadas pelos programadores.

A RAM

Divide-se em 192 páginas de 256 bytes

Endereços úteis aos programado-

res:

Página 0 (endereços \$0000 a \$00FF)

Zonas disponíveis : \$6 a \$9
 \$18 a \$1F
 \$CE e \$CF
 \$D6 e \$D7
 \$EB a \$EF
 \$F9 a \$FF

Os programadores utilizam normalmente estas posições como ponteiros de endereços.

Página 1 (endereço \$0100 a \$01FF)

Zona reservada à pilha (ou stack) do processador.

Esta zona é utilizada nas operações de "PUSH" e "PULL" da pilha

Página 2 (endereço \$0200 a \$02FF)

Zona reservada ao buffer de entrada.

Página 3 (endereço \$0300 a \$3FF)

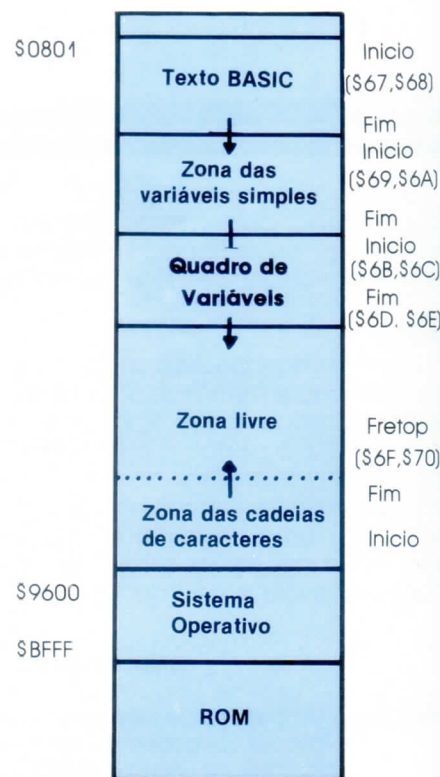
Zona disponível: \$0300 a \$3FF

Esta zona pode servir para implantar pequenos programas inferiores a 208 bytes.

Páginas 4, 5, 6, 7 (endereço \$0400 a \$07FF)

É a memória de ecrã com excepção de 8 bytes reservados às cartas de interface.

Páginas 8 a 192 (endereço \$0800 a \$BFFF)



Fretop: fim da zona livre

As zonas limitadas por ponteiros (ver figura) são de extensão variável de acordo com a importância do seu conteúdo. Por exemplo, se no decurso da programação, o comprimento do texto BASIC aumenta, as zonas das variáveis e dos quadros deslocam-se na direcção dos endereços mais altos.

De igual modo, se, diminuirmos o número de variáveis simples presentes na zona das variáveis, é a zona dos quadros que se desloca para baixo.

Uma simples leitura dos ponteiros correspondentes aos limites (ver a figura) permite-nos situar essas zonas.

A zona das cadeias de caracteres encontra-se no endereço \$9600 se o DOS está implantado e em \$BFFF no caso contrário.

A extensão da zona livre varia em função do comprimento das cadeias de caracteres.



cont. LINGUAGEM MÁQUINA

Nota: A alta resolução gráfica utiliza duas páginas em que a memória de écran cobre os endereços de \$2000 a \$3FFF (página HGR 1) e de \$4000 a \$5FFF (página HGR 2).

Se utilizarmos a alta resolução, deverá ter-se atenção para que a zona "ECRAN" utilizada não seja perturbada por um programa muito longo ou pelas cadeias de caracteres. Depois de ter feito funcionar o programa com um "RUN", basta efectuar uma leitura dos ponteiros das zonas.

Exercício n.º 1:

Os códigos das instruções "INPUT" e "PRINT" são respectivamente 84 (hexa) e BA (hexa).

Enunciado:

Com o auxílio duma sub-rotina "Máquina", substitua todos os "INPUT" por "PRINT"s num programa "BASIC":

Nota: o texto BASIC está implantado na zona "RAM" entre \$801 e o endereço contido no ponteiro (\$69,\$6A).

Solução

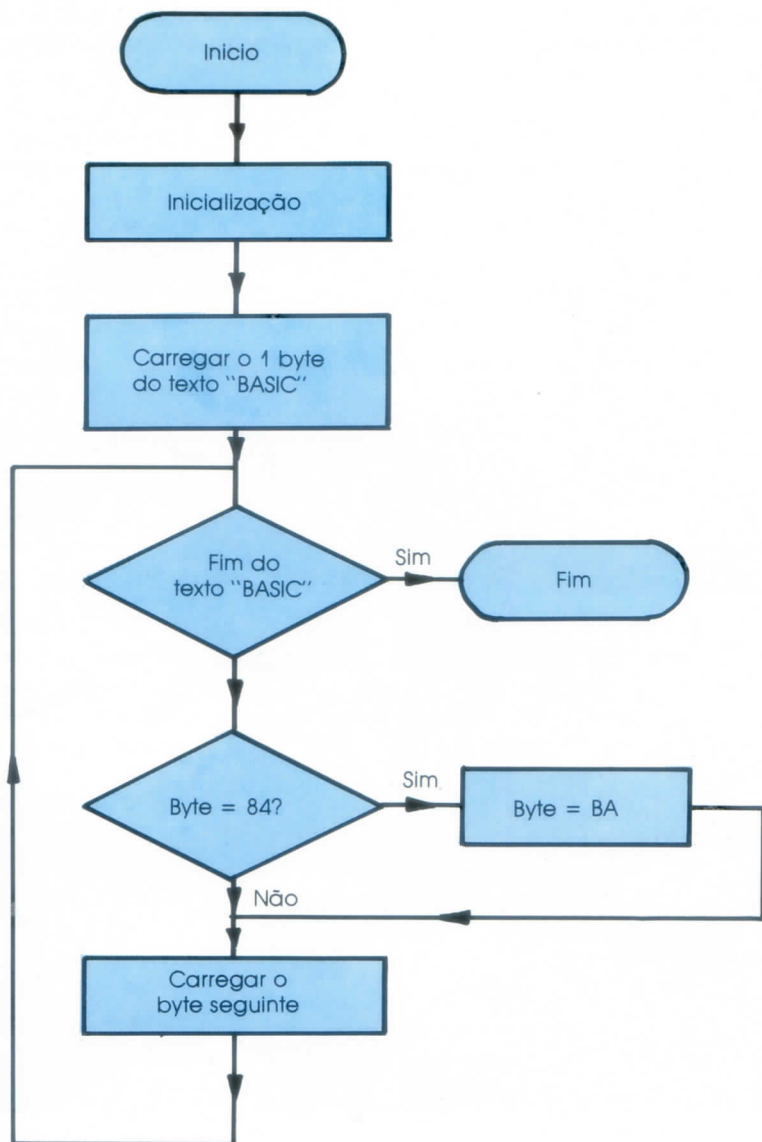
Programa "BASIC" a transformar

Por exemplo:

```
10 HOME
20 INPUT A: INPUT B
30 C=A*B
40 PRINT C
```

Organigrama

Programa "Máquina"



0300-	A9 01	LDA	##01
0302-	8D 0B 03	STA	\$030B
0305-	A9 08	LDA	##08
0307-	8D 0C 03	STA	\$030C
030A-	AD 01 08	LDA	\$0801
030D-	C9 84	CMP	##84
030F-	F0 1D	BEQ	\$032E
0311-	EE 0B 03	INC	\$030B
0314-	D0 03	BNE	\$0319
0316-	EE 0C 03	INC	\$030C
0319-	AE 0B 03	LDX	\$030B
031C-	E4 69	CPX	\$69
031E-	F0 03	BEQ	\$0323
0320-	4C 0A 03	JMP	\$030A
0323-	AE 0C 03	LDX	\$030C
0326-	E4 6A	CPX	\$6A
0328-	F0 03	BEQ	\$032D
032A-	4C 0A 03	JMP	\$030A
032D-	60	RTS	
032E-	AC 0B 03	LDY	\$030B
0331-	8C 3D 03	STY	\$033D
0334-	AC 0C 03	LDY	\$030C
0337-	8C 3E 03	STY	\$033E
033A-	A9 BA	LDA	##BA
033C-	8D 00 00	STA	\$0000
033F-	EE 0B 03	INC	\$030B
0342-	D0 03	BNE	\$0347
0344-	EE 0C 03	INC	\$030C
0347-	4C 0A 03	JMP	\$030A

Depois de ter introduzido os dois programas em "RAM", digite em modo directo "CALL 768" seguido de "RETURN".

Através de um "LIST", pode constatar que todos os "INPUT" passaram a "PRINT".

CLUBE COMMODORE

Coordenador:
Sadik Jamal
Rua da Restauração 83, 2.º
4000 Porto
Telef. 699382; Telex 23156



CLUBE DE SOFTWARE de Reguengos de Monsaraz

Coordenadores:
José Gabriel Calisto e
Maria da Encarnação Carvalho
Sítio da Cartuxa, n.º 1 — 1.º Andar
7200 Reguengos de Monsaraz

CLUBE DE SOFTWARE da Bélgica «Leosoft»

École Industrielle Leo Collard
(Mons-Belgique)
Coordenador:
Philippe Laurent
Telef. 00/32/65/335948 (6.ª Feiras
das 16 às 20 h)
10 Boulevard Kennedy
7000 Mons Belgique

CLUBE DE SOFTWARE da Região de LISBOA

Coordenador:
João Manuel Pereira Salgueiro
Estrada de Benfica, 518, 3.ª-Dt.º
1500 Lisboa
Tel. 7142007

CLUBE ZX Vila Pouca de Aguiar

Coordenador:
Paulo Jorge Santos
Praça do Comércio — Café Califa
5450 Vila Pouca de Aguiar

CLUBE DE SOFTWARE da Região de COIMBRA

Coordenador:
António Luís Alves
Av. Dr. Veiga Simão, 40
3530 MANGUALDE
Tel. 62434
TLX: 53538 ATT — A.L. ALVES

CLUBE DE SOFTWARE da Região de PORTIMÃO

Coordenador:
José António da Mata P. Raposo
Rua Heróis da Restauração, n.º 24
Dt.º
8500 PORTIMÃO
Tel. 24173

Fazemos:

Decoração de interiores e exteriores;
serviços de manutenção; floreiras
decorativas; entregas ao domicílio.

Eis alguns dos nossos clientes:

Presidência da República, B.P.S.M.
(todas as dependências de Lisboa),
Hotéis (Tivoli, Penta, Meridien, etc.),
Centro de Arte Moderna, Centro
Comercial das Amoreiras...

Somos:



HORTO DO CAMPO GRANDE

(Junto à Alameda da Universidade)

Telefs: 772383/732195

e agora no

Centro Comercial das Amoreiras, loja
1007

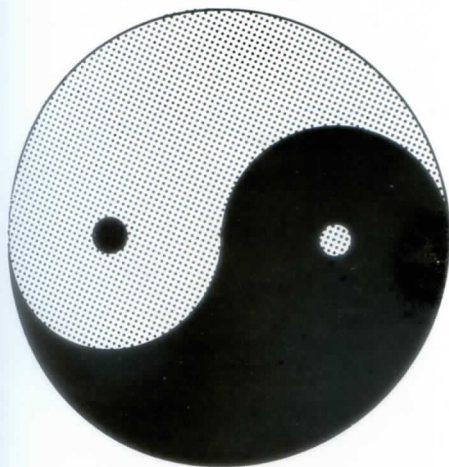




FORÇAS UNIVERSAIS

M. ABRASSART

(Yinn e Yang)



A teoria da acupunctura vê em todos os fenômenos, o resultado da luta de duas forças opostas, (reversíveis uma na outra), cada uma delas sendo por assim dizer o reflexo simétrico da sua contrária.

Essas forças são o Yinn e o Yang.

O Yinn (em negro) é o negativo: as trevas, o frio, a inércia...

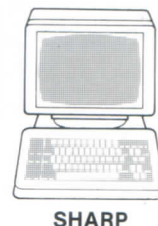
A natureza do Homem, não é mais do que uma das formas desta oposição entre o Yinn e o Yang, e todas as doenças têm a sua origem num desequilíbrio en-

tre estas duas forças.

Yinn e o Yang são antagônicos mas inseparáveis. Assim, toda a qualidade exterior evidente não é mais que o reforço da sua antagônica interior; o Yinn vê nascer no seu seio o jovem Yang, simetricamente o Yinn nasce no Yang.

O acto terapêutico consiste então em picar o meridiano, segundo técnicas particulares, afim de obter sobre ele um efeito Yinn, ou um efeito Yang, isto é a "dispersá-lo" ou "tonificá-lo".

<pre> G:REM YINN-YANG 10: CLEAR: GRAPH: GOTO 45 15: COLOR C 20: FOR Z=ZL TO ZH STEP 200/R 25: X=R*SIN Z: Y=R*COS Z 30: IF Z=0 OR Z=180 THEN GLCURSOR(X,Y): GOTO 40 35: LINE -(X,Y) 40: NEXT Z: RETURN 45: GLCURSOR (110,-125): SORGN 50: R=100: ZH=180: GOSUB 15 </pre>	<pre> 55: ZL=ZH: ZH=360: C=3: GOSUB 15 60: ZH=ZL: ZL=0: R=R/2: GLCURSOR (O,R): SORGN: GOSUB 15 65: ZL=ZH: ZH=360: C=0: GLCURSOR (O,-2*R): SORGN: GOSUB 15 70: ZL=0: R=R/5: C=3: GOSUB 15 75: GLCURSOR (O,10*R): SORGN: C=0: GOSUB 15 80: GLCURSOR (-10*R, -25*R): TEXT: END </pre>
---	--



SHARP



Vende-se Máquina de contabilidade marca Hermes mod.º F-3
Preço 60.000\$00
Telefone 056-24531

Amstrad CPC-464 procuro possuidores ou futuros para troca de programas e ideias, possível formação do Clube Amstrad, contactar: João Amaral - Telef. 895270

Vende-se Sharp MZ 700, a estrear, 50 contos
Telef. 675295

PEDRO MIGUEL FIGUEIREDO

Desejo trocar ideias sobre linguagens do computador. Contactar, por carta, para:
PEDRO MIGUEL FIGUEIREDO
R. Dr. Aníbal Beleza, 243-1.º-Dt.º
3720 OLIVEIRA DE AZEMÉIS

Utilizador de Commodore 64 procura contactos para troca de ideias e software. Precisa bom programa de ficheiro. Contacta José Fernandes, rua do Algarve 71; 7700 Almodovar.

Registo Dados
Operação Minicomputadores

Aceitam-se tarefas ou prestação de serviços. Regime de turnos. Contactar: Qt.º St.º Amaro, Imp. IX n.º 2 - 8.º A — Laranjeiro



4 EM LINHA

4 em linha: um jogo repousante que vai divertir as pessoas dos 7 aos 77 anos.

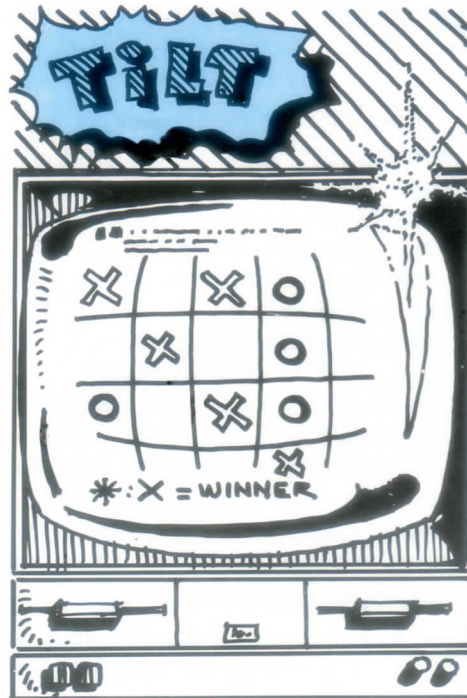
O objectivo do jogo é muito simples: para ganhar basta alinhar 4 peças numa grelha com 7 casas de comprimento por 6 de largura.

Podemos alinhá-los na horizontal, vertical ou diagonal. O primeiro a consegui-lo é o vencedor. Para mais esclarecimentos consulte o seu programa.

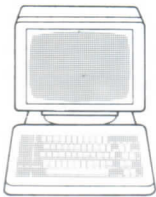


Nota:

No programa devem-se substituir os números e as letras enquadradas pelas teclas correspondentes do quadro que se segue.



1	: SHIFT+HOME
2	: CTRL+2
3	: HOME (S em inverso)
4	: ↓(Q em inverso)
5	: CTRL+9 (R em inverso)
6	: CTRL+8 (I em inverso)
7	: CTRL+0
8	: CTRL+4
9	:TECLA COMMODORE+3
10	: →
11	: ↑
12	: CTRL+3
13	: CTRL+7
14	: CTRL+1
15	: SHIFT+Q
A	: SHIFT+U
B	: SHIFT+B
C	: SHIFT+C
D	: SHIFT+I
E	: SHIFT+K
F	: SHIFT+J



COMMODORE

```

1 REM *****
2 REM *PROGRAMA ESCRITO POR *
3 REM ** PHILIPPE POURTOIS **
4 REM *****
10 CLR:GOTO 7000
100 PRINT " 1 2 ":POKE
53280,1:POKE 53281,5
110 FOR I=1063 TO 1024 STEP
-1:POKE I,160:POKE 54272+I,7:
NEXT
120 FOR I=1064 TO 1984 STEP
40:POKE I,160:POKE 54272+I,7:
NEXT
130 FOR I=1985 TO 2023:POKE
I,160:POKE 54272+I,7:NEXT
140 FOR I=1983 TO 1103 STEP
-40:POKE I,160:POKE 54272+I,7
NEXT
150 PRINT " 3 4 4 5 6 7 2
4 EM LINHA"
160 PRINT " 5 6 7 2 -----"
170 PRINT " 4 5 6 7 2 O JOGO
CONSIESTE EM ALINHAR 4 PEDES
NUMA"
175 PRINT " 4 5 6 7 2 GRELHA
DE 7 CASAS DE COMPRIMENTO "
176 PRINT " 4 5 6 7 2 POR 6
CASAS DE LARGURA PARA GANHAR"
180 PRINT " 4 5 6 7 2 PODE-SE
ALINHAR HORIZONTALMENTE,"
190 PRINT " 4 5 6 7 2
VERTICALMENTE OU EM DIAGONAL"
200 PRINT " 4 5 6 7 2 O
PRIMEIRO QUE CONSEGUIR E O
VENCEDOR"
210 PRINT " 4 5 6 2 CARREGUE
NUMA TECLA"
220 GET R$:IF R$="" THEN 220
500 PRINT " 2 T 1 ":POKE
53280,10:POKE 53281,11:DR=1:
TR=0:CD=0
510 FOR X=1315 TO 1755 STEP
40:POKE X,66:NEXT X:FOR X=
1317 TO 1757 STEP 40:POKE
X,66:NEXT X
520 FOR X=1319 TO 1759 STEP
40:POKE X,66:NEXT X:FOR X=
1321 TO 1761 STEP 40:POKE X,
66:NEXT X
530 FOR X=1323 TO 1763 STEP
40:POKE X,66:NEXT X:FOR X=
1325 TO 1765 STEP 40:POKE X,
66:NEXT X
540 FOR X=1327 TO 1767 STEP
40:POKE X,66:NEXT X:FOR X=
1329 TO 1769 STEP 40:POKE X,
66:NEXT X
550 FOR X=1795 TO 1809:POKE
X,119:NEXT X
560 FOR X=1315 TO 1328 STEP
2:POKE X,67:NEXT X:FOR Y=1396
TO 1408 STEP 2:POKE X,67:
NEXT X
570 FOR X=1475 TO 1488 STEP

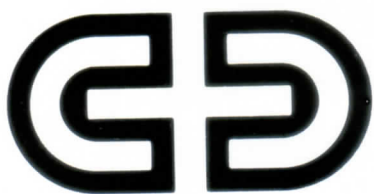
```


SÉRIE 800
A geração de
médios e grandes
computadores

**MICRO
PLATO**
A formação
individualizada

**SUPORTES
MAGNÉTICOS**
Qualidade
e tecnologia
no tratamento
da informação

**SERVIÇOS
DE
CONSULTORIA**
O suporte
indispensável



**CONTROL
DATA**

SOLUÇÕES PARA TODOS OS PROBLEMAS DE INFORMÁTICA

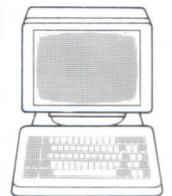
CONTROL DATA PORTUGUESA, S.A.R.L

Av. Eng.º Duarte Pacheco — Empreendimento das Amoreiras
Torre 2 — 15.º Piso — 1200 Lisboa
Telefs.: 693700/693747/693806/693723
Telex: 18479 Lisdat P



cont. 4 EM LINHA

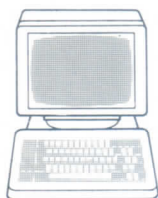
1490 C=1	1930 IF D=X THEN 2100
1500 IF PEEK (K)=87 AND PEEK (K-80)=87 THEN 1560	1940 NEXT X
1510 E=1	1950 FOR X=1362 TO 1608 STEP 82
1520 IF C)=4 OR E)=4 THEN 6000	1960 IF D=X THEN 2110
1530 NEXT K	1970 NEXT X
1540 GOTO 1600	2000 N=1440:M=1596:GOTO 3000
1550 C=C+1:GOTO 1520	2010 N=1442:M=1676:GOTO 3000
1560 E=E+1:GOTO 1520	2020 N=1444:M=1756:GOTO 3000
1600 D=A:C=1:E=1:O=78	2030 N=1446:M=1758:GOTO 3000
1610 FOR X=1362 TO 1596 STEP 78	2040 N=1526:M=1760:GOTO 3000
1620 IF D=X THEN 2000	2050 N=1606:M=1762:GOTO 3000
1630 NEXT X	2060 N=1598:M=1762:GOTO 3000
1640 FOR X=1364 TO 1676 STEP 78	2070 N=1518:M=1764:GOTO 3000
1650 IF D=X THEN 2010	2080 N=1438:M=1766:GOTO 3000
1660 NEXT X	2090 N=1440:M=1768:GOTO 3000
1670 FOR X=1366 TO 1756 STEP 78	2100 N=1442:M=1688:GOTO 3000
1680 IF D=X THEN 2020	2110 N=1444:M=1608:GOTO 3000
1690 NEXT X	3000 D=A:C=1:E=1
1700 FOR X=1368 TO 1758 STEP 78	3005 FOR K=N TO M STEP 0
1710 IF D=X THEN 2030	3010 IF PEEK (K)=81 AND PEEK (K-0)=81 THEN 3080
1720 NEXT X	3020 C=1
1730 FOR X=1448 TO 1760 STEP 78	3030 IF PEEK (K)=87 AND PEEK (K-0)=87 THEN 3090
1740 IF D=X THEN 2040	3040 E=1
1750 NEXT X	3050 IF C)=4 OR E)=4 THEN 6000
1760 FOR X=1528 TO 1762 STEP 78	3060 NEXT K
1770 IF D=X THEN 2050	3070 IF TR=0 THEN 1790
1780 NEXT X	3075 GOTO 820
1790 D=A:O=82:TR=1	3080 C=C+1:GOTO 3050
1800 FOR X=1516 TO 1762 STEP 82	3090 E=E+1:GOTO 3050
1810 IF D=X THEN 2060	6000 PRINT " 11 11 5 7 "
1820 NEXT X	6010 PRINT " 11 5 7 0 "
1830 FOR X=1436 TO 1764 STEP 82	JOGADOR" Z "GANHOU"
1840 IF D=X THEN 2070	6100 S=54272
1850 NEXT X	6110 FOR L=0 TO 24:POKE S+L, 0:NEXT
1860 FOR X=1356 TO 1766 STEP 82	6120 POKE S+14,5
1870 IF D=X THEN 2080	6130 POKE S+18,16
1880 NEXT X	6140 POKE S+3,1
1890 FOR X=1358 TO 1768 STEP 82	6150 POKE S+24,143
1900 IF D=X THEN 2090	6160 POKE S+6,240
1910 NEXT X	6170 POKE S+4,65
1920 FOR X=1360 TO 1688 STEP 82	6180 FR=5389:O=3:P=4
	6190 FOR T=1 TO 200
	6195 IF INT(T/20)=T/20 THEN O=O+1
	6200 FQ=FR+PEEK (S+27)*3.5
	6210 HF=INT(FQ/256):LF=FQ-HF*256
	6220 POKE S+0,LF:POKE S+1,HF
	6225 IF INT(T/20)=T/20 THEN POKE 53280,0



COMMODORE



cont. 4 EM LINHA



COMMODORE

```

6226 POKE 53281,11
6230 NEXT T
6240 POKE S+24,0
6245 POKE 53280,10:POKE 53281
,11
6250 GOTO 6250
7000 PRINT " 2 T 1 ":POKE
53280,1:POKE 53281,1
7002 PRINT " 3 4 4 4 4 4 4
4 4 4 4 4 10 10 10 10 12 ***
13 5 UM POUÇO DE PACIENCIA
S.F.F. 7 12 ***"
7009 S=54272:FOR L=S TO S+24:
POKE L,0:NEXT
7010 DIM H(2,200),L(2,200),
C(2,200)
7020 DIM FQ(11)
7025 V(0)=17:V(1)=65:V(2)=33
7030 POKE S+10,8:POKE S+22,
128:POKE S+23,244
7040 FOR I=0 TO 11 :READ
FQ(I):NEXT
7100 FOR K=0 TO 2
7110 I=0
7120 READ NM
7130 IF NM=0 THEN 7250
7140 WA=V(K):WB=WA-1:IF NM<0
THEN NM=NM:WA=0:WB=0
7150 DR%=NM/128:OC%=(NM-128*
DR%)/16
7160 NT=NM-128*DR%-16*OC%
7170 FR=FQ(NT)
7180 IF OC%=7 THEN 7200
7190 FOR J=6 TO OC% STEP -1:
FR=FR/2:NEXT J
7200 HF%=FR/256:LF%=FR-256*
HF%
7210 IF DR%=1 THEN H(K,I)=HF%
:L(K,I)=LF%:C(K,I)=WA:I=I+1:
GOTO 7120
7220 FOR J=1 TO DR%-1
7221 H(K,I)=HF%
7222 L(K,I)=LF%
7223 C(K,I)=WA
7224 I=I+1
7225 NEXT
7230 H(K,I)=HF%:L(K,I)=LF%:
C(K,I)=WB
7240 I=I+1:GOTO 7120
7250 IF I>IM THEN IM=I
7260 NEXT
7280 PRINT " 14 T 1 ":POKE
53280,0:POKE 53281,0
7290 PRINT " 3 4 4 4 10 10 10
10 10 10 10 10 5 8 SOFTWARE"
7292 PRINT " 10 10 10 10 10
10 10 10 5 7 5 "
7294 PRINT " 10 10 10 10 10
10 10 10 5 7 2 A C C C C C C
C C C C C C C C C C 5 8 "
7296 PRINT " 10 10 10 10 10
10 10 10 5 7 2 B B 5 8 "
7300 PRINT " 10 10 10 10 10
10 10 10 5 L 7 2 B 5 9
APRESENTA 7 2 B 5 8 L "
7310 PRINT " 10 10 10 10 10
10 10 10 5 E 7 2 B B 5 8
E "
7320 PRINT " 10 10 10 10 10
10 10 10 5 7 2 F C C C C C C
C C C C C C C C C C 5 8 "
7330 PRINT " 10 10 10 10 10
10 10 10 5 L 7 2 B 5 8 4
EM LINHA 7 2 B 5 8 L "
7340 PRINT " 10 10 10 10 10
10 10 10 5 0 7 2 B A C C C D
B 5 8 0 "
7350 PRINT " 10 10 10 10 10
10 10 10 5 G 7 2 B B B B
5 8 G "
7355 PRINT " 10 10 10 10 10
10 10 10 5 I 7 2 B B B
B 5 8 I "
7360 PRINT " 10 10 10 10 10
10 10 10 5 C 7 2 B F C C C E
B 5 8 C "
7365 PRINT " 10 10 10 10 10
10 10 10 5 I 7 2 B A D 1 5
B 5 8 I "
7370 PRINT " 10 10 10 10 10
10 10 10 5 E 7 2 B B B B
5 8 E "
7375 PRINT " 10 10 10 10 10
10 10 10 5 L 7 2 B B B B
5 8 L "
7380 PRINT " 10 10 10 10 10
10 10 10 5 7 2 B F E B 5
8 "
7390 PRINT " 10 10 10 10 10
10 10 10 5 7 2 F C C C C C C
C C C C C C C C C C 5 8 "
7395 PRINT " 10 10 10 10 10
10 10 10 5 7 5 "
7400 PRINT " 10 10 10 10 10
10 10 10 5 SOFTWARE 7 "
7500 POKE S+5,0:POKE S+6,240
7510 POKE S+12,85:POKE S+13
,133
7520 POKE S+19,10:POKE S+20,
197
    
```

commodore

64



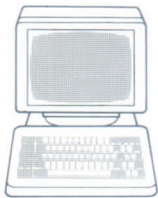
COMERCIAL LABORUM
REPRESENTANTE EXCLUSIVO

Rua da Restauração 83-2 4000 PORTO
405767, 400382

Rua Arco do Carvalho 59-6 Dto 1000 LISBOA
450793



cont. 4 EM LINHA



COMMODORE

7530 POKE S+24, 31	8070 DATA 329, 328, 1609, 578,
7540 FOR I=0 TO IM	834
7550 POKE S, L(0, I):POKE S+7,	8080 DATA 324, 322, 327, 585,
L(1, I):POKE S+14, L(2, I)	1602
7560 POKE S+1, H(0, I):POKE S+8	8999 DATA 0
H(1, I):POKE S+15, H(2, I)	9000 DATA 567, 566, 567, 304, 306
7570 POKE S+4, C(0, I):POKE	, 308, 310
S+11, C(1, I):POKE S+18, C(2, I)	9010 DATA 1591, 567, 311, 310,
7580 FOR T=1 TO 80:NEXT:NEXT	567
7590 FOR T=1 TO 200:NEXT:POKE	9020 DATA 306, 304, 299, 308
S+24, 0	9030 DATA 304, 171, 176, 306, 291
7600 DATA 34334, 36376, 38539,	, 551, 306, 308
40830	9040 DATA 310, 308, 310, 306, 295
7610 DATA 43258, 45830, 48556,	, 297, 299, 304
51443	9050 DATA 1586, 562, 567, 310,
7620 DATA 54502, 57743, 61176,	315, 311
64814	9060 DATA 308, 313, 297
7999 DATA 0	9070 DATA 1586, 567, 560, 311,
8000 DATA 583, 585, 583, 583,	309
327, 329	9090 DATA 1577, 299, 295, 306,
8010 DATA 1611, 583, 585, 578,	310, 311, 304
578, 578	9100 DATA 562, 546, 1575
8020 DATA 196, 198, 583, 326, 578	9200 DATA 0
8030 DATA 326, 327, 329, 327, 329	9300 GOTO 100
, 326, 578, 583	10000 PRINT " 11 11 5 7
8040 DATA 1606, 582, 322, 324,	"
582, 587	10010 PRINT " 11 5 7 EMPATE"
8050 DATA 329, 327, 1606, 583	10020 GOTO 6100
8060 DATA 327, 329, 587, 331, 329	



Nota da Redacção

Por absoluta falta de espaço vimo-nos na necessidade de sacrificar neste número a rubrica Correio.

Com os pedidos de desculpa devidos em especial aos leitores que nos escreveram, prometemos na medida do possível reparar esta falta na próxima edição.

Pedimos aos leitores que desejarem renovar a assinatura, que o façam enquanto esta esteja em vigor, para facilitar os nossos serviços administrativos e também para poderem conservar o mesmo número de assinante.

Aproveitamos ainda para informar os nossos leitores da região do Porto, que vamos estar presentes com um stand na Inforpor 85 (Exposição de Tecnologias de Informação, de 5 a 8 de Dezembro no Palácio de Cristal), onde poderão adquirir qualquer número da Software, bem como capas para encadernar as revistas (iguais às que oferecemos com o nosso número de aniversário).

Esperamos a sua visita!

Convites

Alguns dos nossos assinantes (esperamos fazê-lo para todos os da zona do Porto) vão receber em casa convites que lhe facultam a entrada na Inforpor 85.

Uma pequena atenção que esperamos repetir em futuras ocasiões. Só para assinantes!



Acordo para Padrão Europeu de Computadores

Thomson, o grupo francês de electrónica, juntamente com a Olivetti e a Acorn, a companhia dos computadores pessoais controlada pelos italianos, estudam a possibilidade de desenvolver um computador pessoal com fins domésticos e educacionais.

Na base deste possível acordo está a tentativa de criar um padrão europeu para competir com os padrões japoneses e americanos.



As perspectivas são, além da comercialização dos produtos existentes, fazer-se uma eventual produção em conjunto.

Não é de agora que a Thomson procura um sócio europeu, tentou antes uma ligação com a Philips, no intuito de obter apoio a sua recente entrada no mercado dos computadores pessoais.

Nixdorf Gold Cup

Consciente das suas responsabilidades sociais, como maior empresa nacional na sua área de actividade, a RIMA dedica desde há vários anos especial atenção à promoção do Golf.

Tendo encontrado total apoio por parte do Oporto Golf Club (terceiro mais antigo clube de golf da Europa Continental), a RIMA reali-



zou, em conjunto com a sua representada, a "Nixdorf Gold Cup".

De salientar o principal troféu em jogo, a "Nixdorf Gold Cup", única taça de ouro disputada no nosso país e uma das poucas existentes na Europa, o que constituiu um dos grandes aliciantes do principal torneio de golfe amador em Portugal.



Logiser

Nova associada da Profabril

Numa aposta clara na modernização e dinamização da indústria de "software", a Profabril acaba de adquirir 49% da Logiser, uma das primeiras softwarehouses portuguesas.

Esta empresa de alta tecnologia foi criada por iniciativa de um grupo de professores universitários, visando responder de uma forma integrada a um conjunto de problemas e necessidades nos meios empresariais na produção de "software".



NOVO PACKAGE PARA FARMÁCIAS

A **Olivetti Holanda**, em colaboração com a **Euromed**, reputada Software House holandesa, desenvolveu um "package" — **APCOS** — para a sua linha de equipamentos de médio/grande porte, a **L1-MOS**, com o objectivo da informatização integral da Farmácia Oficina holandesa.

O sucesso desta solução é, hoje, bem afirmado quer pelo seu reconhecimento oficial por parte da **SIGN**, instituição nacional das farmácias holandesas, quer pela sua instalação em já cerca de 80 farmácias, em pouco mais de 1 (um) ano de existência.

Commodore AMIGA

O AMIGA já existe. De aspecto clássico, bem PC, o seu interior é uma melhoria substancial em relação ao Macintosh. Em vez de seguir a linha do Atari e adoptar o GEM (GRAPHICS ENVIRONMENT MANAGER) um programa que



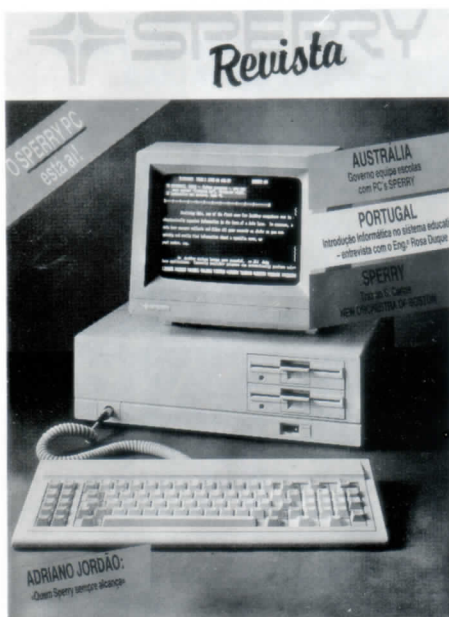
obriga o 68000 a fazer todo o trabalho, a Commodore desenvolveu circuitos especializados que o libertam:

- Circuito de acesso à memória e animação gráfica
- circuito de gestão de ecrã
- circuito integrador estereo

Como exemplo das incríveis potencialidades do Amiga, podemos dizer que este é capaz de aceder ao ecrã com uma velocidade de 1 milhão de pixels por segundo, uma definição de 600x400 com 32 cores simultâneas escolhidas de 4096 diferentes.

É provável que o Amiga seja comercializado já a partir de Janeiro, e por um preço na ordem dos 400 contos!

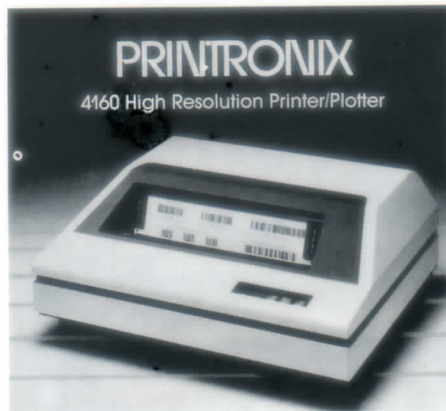
Esteja atento ao número de Janeiro, onde faremos uma análise detalhada do que vai ser provavelmente a revelação do início de 86.



Novas Publicações

Queremos dar as Boas Vindas a dois novos "colegas" da informação, a revista Sperry e a revista da Landry.

Trimestral e de melhor apresentação a primeira, mensal e sem grandes cuidados do aspecto gráfico a segunda, são ambas de distribuição gratuita.



Printronix

A Printronix, um dos maiores construtores mundiais, vai instalar-se em Portugal.

Jean Marie Pirnay, Director Geral para a Europa, esteve recentemente entre nós para divulgar os projectos daquela companhia americana.

Seleccionar dois representantes, um que estará especialmente vocacionado para os contactos com África, é uma das prioridades.

Nesta conferência de imprensa realizada na Sociedade 2000, falou-se ainda das impressoras de código de barras, um dos primeiros produtos a serem comercializados.

Control Data

A Control Data Corporation recebeu quase em simultâneo três novas encomendas de sistemas de processamento multiparalelo Cyberplus e respectivos Cybers da serie 180, num negócio que envolve cerca de seis milhões de dólares.

Os processadores Cyberplus, ligados a computadores Cyber,

multiplicam significativamente a capacidade de processamento de cada computador. Atingem-se desta forma velocidades de computação, antes restritas a supercomputadores, a menor preço.

... também na China

No seguimento de relações comerciais que tiveram início em 1974 com o fornecimento de dois computadores Cyber ao Ministério do Petróleo, a Control Data Corporation acaba de firmar um contrato no valor de 30 milhões de dólares para fornecimento de computadores e equipamentos ao mesmo Ministério da República Popular da China.

O material a fornecer inclui cinco sistemas Cyber 180 e software de apoio para processamento de dados sísmicos, simulação de reservatórios, construção à superfície e gestão de campos petrolíferos. Incluído na venda está o programa de software Geomaster, para exploração petrolífera.

...e com Novas Instalações em Lisboa

A Control Data Portuguesa, empresa subsidiária da Control Data Corporation, devido ao alargamento dos seus quadros de pessoal e de infraestruturas, bem como a fim de permitir um mais fácil contacto com os seus clientes e fornecedores, centralizou todos os seus Departamentos e Serviços em novas instalações, sitas na Av.º Eng.º Duarte Pacheco, Empreendimento das Amoreiras. Torre 2 — 15.º Piso.

Dentro de dois meses na Software...

- A reportagem exhaustiva da INFOPOR 85
- O programa 4 em linha para o Spectrum
- "Secção Dourada", um programa de Sérgio Granhão que permite calcular o máximo UMAX de uma função genérica U(X)
- Commodore Amiga, a revelação!

isto e muito, muito mais!



PARA ESTAR "À CORRENTE"

E. DORMONT

...circuito ressonante RLC paralelo...

Se não encontrou dificuldades na teoria relativa ao circuito ressonante RLC série que apresentámos no último número, não hesite em lançar mãos à obra e tome conhecimento agora, do circuito ressonante RLC paralelo.



Não há grande diferença teórica, salvo o facto da selectividade se inverter em relação ao circuito em série.

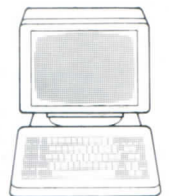
Com efeito:

$$Ndb = \sqrt{1 + 25Q^2}$$

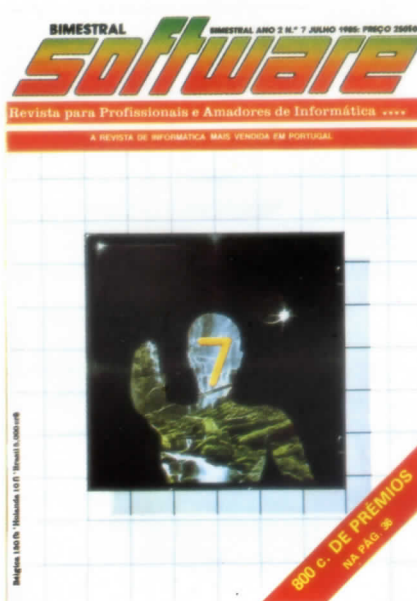
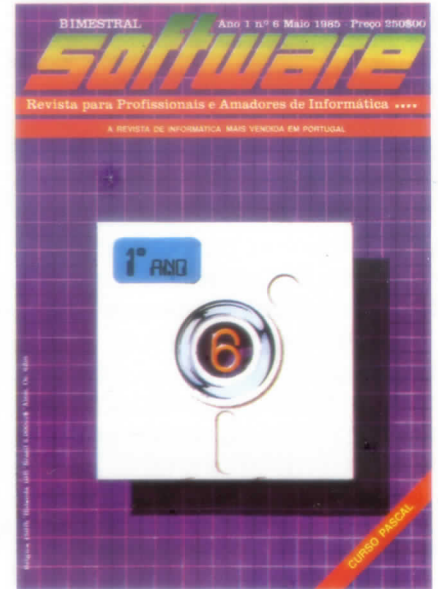
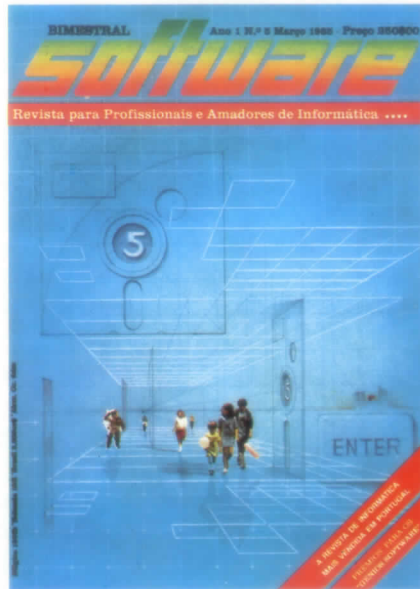
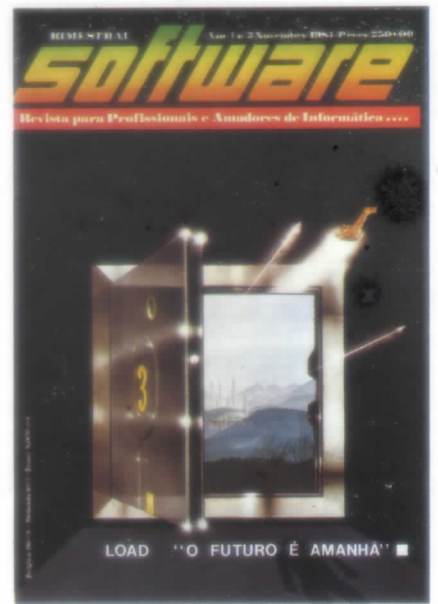
```

10 ZZ$=CHR$(4)
20 REM CALCULO DUM CIRCUITO
RESSONANTE PARALELO
30 REM AUTOR DORMONT ERIC
40 REM COPYRIGHT SOFTWARE E O
AUTOR
50 REM
60 DIM S(1000)
70 A$="*CALCULO DUM CIRCUITO
RLC RESSONANTE*"
80 AA$="*****
*****"
90 HOME
100 REM ENTRADA DOS VALORES
R,L,C
110 VTAB 2:PRINT A$
120 VTAB 3:PRINT AA$
130 VTAB 5:INVERSE:PRINT "
INTRODUZA O VALOR DOS
COMPONENTES R,L,C":NORMAL
140 VTAB 8:PRINT "1.
RESISTENCIA EM OHM...:"
150 VTAB 10:PRINT "2.
INDUTANCIA EM MICROHENRY...:"
160 VTAB 12:PRINT "3.
CAPACIDADE EM PICOFARAD:"
170 REM
180 VTAB 8:HTAB 27:INPUT R:IF
K=10 THEN 230
190 VTAB 10:HTAB 27:INPUT LL:
IF K=10 THEN 230
200 VTAB 12:HTAB 27:INPUT CC:
IF K=10 THEN 230
210 VTAB 18:PRINT "PARA
MODIFICAR UMA RUBRICA,
CARREGUE"
220 VTAB 20:PRINT "O SEU
NUMERO OU <F> PARA ACABAR."
230 VTAB 20:HTAB 31:GET Q1$:
K=10
240 IF Q1$="1" THEN 180
250 IF Q1$="2" THEN 190
260 IF Q1$="3" THEN 200
270 IF Q1$="F" THEN 300
280 GOTO 230
290 REM CALCULOS DO PROBLEMA
300 C=CC*(10↑(-12))
310 L=LL*(10↑(-6))
320 PI=3.14159265
330 FO=(1/(2*PI))*(((1/(L*C))
-((R↑2)/(L↑2)))↑0.5)
340 W0=2*PI*FO
350 Q0=(L*W0)/R
360 B=FO/Q0
370 R$=STR$(R)
380 L$=STR$(LL)
390 C$=STR$(CC)
400 Q0$=STR$(Q0)
410 FO$=STR$(FO/1000)
420 B$=STR$(B/1000)
430 W0$=STR$(W0)
440 ER$="R="+R$+" OHM"
450 EL$="L="+L$+" MICROHENRY"
460 EC$="C="+C$+" PICOFARAD"
470 EQ0$="Q NA RESSONANCIA="
+Q0$
480 EFO$="FREQUENCIA DE
RESSONANCIA="+FO$+" KHz"
490 EB$="BANDA PASSANTE A
-3Db="+B$+" KHz"
500 EWO$="W NA RESSONANCIA="
+W0$
510 ER=LEN(ER$):ER=(40-(INT
(ER)))/2
520 EL=LEN(EL$):EL=(40-(INT
(EL)))/2
530 EC=LEN(EC$):EC=(40-(INT
(EC)))/2
540 EQ0=LEN(EQ0$):EQ0=(40-
(INT(EQ0)))/2
550 EFO=LEN(EFO$):EFO=(40-
(INT(EFO)))/2
560 EB=LEN(EB$):EB=(40-(INT
(EB)))/2

```



APPLE II



FAÇA-SE ASSINANTE. DESCUBRA A RAZÃO PORQUE SOMOS A REVISTA DE INFORMÁTICA MAIS VENDIDA EM PORTUGAL.

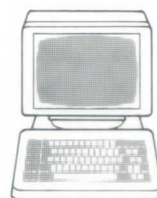


cont. PARA ESTAR "À CORRENTE"

```

570 EW0=LEN(EW0$):EW0=(40-
(INT(EW0)))/2
580 REM ENTRADA DA GAMA DE
FREQUENCIAS
590 HOME
600 VTAB 1:PRINT A$
610 VTAB 2:PRINT AA$
620 INVERSE
630 VTAB 4:HTAB ER:PRINT ER$
640 VTAB 6:HTAB EL:PRINT EL$
650 VTAB 8:HTAB EC:PRINT EC$
660 VTAB 10:HTAB EF0:PRINT
EF0$
670 VTAB 12:HTAB EW0:PRINT
EW0$
680 VTAB 14:HTAB EQ0:PRINT
EQ0$
690 VTAB 16:HTAB EB:PRINT EB$
700 NORMAL
710 VTAB 17:PRINT "1.
FREQUENCIA MINIMA:"
720 VTAB 19:PRINT "2.
FREQUENCIA MAXIMA:"
730 VTAB 21:PRINT "3. SALTO DE
FREQUENCIA:"
740 HTAB 7:VTAB 22:FLASH:
PRINT "AS FREQUENCIAS EM Hz
S.F.F.!" :NORMAL
750 VTAB 17:HTAB 22:INPUT FI:
IF K=20 THEN 790
760 VTAB 19:HTAB 22:INPUT FX:
IF K=20 THEN 790
770 VTAB 21:HTAB 22:INPUT SF:
IF K=20 THEN 790
780 VTAB 23:PRINT "PARA
MODIFICAR (1,2,3) OU <F> PARA
ACABAR"
790 VTAB 23:HTAB 38:GET Q2$:
K=20
800 IF Q2$="1" THEN 750
810 IF Q2$="2" THEN 760
820 IF Q2$="3" THEN 770
830 IF Q2$="F" THEN 860
840 GOTO 790
850 REM ESCOLHA DO MODO DE
CALCULO DE SELECTIVIDADE
860 HOME
870 VTAB 2:PRINT A$
880 VTAB 3:PRINT AA$
890 VTAB 7:PRINT "ESCOLHE UMA
RESOLUCAO"
900 VTAB 9:PRINT "DO FACTOR
DE SOBRETENSAO:"
910 VTAB 15:HTAB 10:PRINT "1.
EM DECIBEL"
920 VTAB 18:HTAB 10:PRINT "2.
RELACAO I/I0"
930 VTAB 22:PRINT "ESCOLHA 1
OU 2"
940 GET Q3
950 IF Q3=1 THEN 990
960 IF Q3=2 THEN 990
970 GOTO 940
980 REM CALCULO DA
SELECTIVIDADE
990 HOME
1000 FOR F=FI TO FX STEP SF
1010 W=2*PI*F
1020 D=(F-F0)/F0
1030 A=A+1
1040 Q=(L*W)/R
1050 ON Q3 GOSUB 15000,10000
1060 S(A)=S
1070 HTAB 20:VTAB 12:INVERSE:
PRINT A
1080 NORMAL
1090 NEXT F
1100 HOME
1110 REM ESCOLHA DO MODO DE
IMPRESSAO
1120 VTAB 4:HTAB 16:PRINT
"*MENU.1*"
1130 VTAB 5:HTAB 16:PRINT
"====="
1140 VTAB 10:HTAB 10:PRINT
"1.NA IMPRESSORA"
1150 VTAB 16:HTAB 10:PRINT
"2.NO ECRAN"
1160 VTAB 22:INVERSE:PRINT
"A SUA ESCOLHA ":NORMAL
1170 VTAB 22:HTAB 22:GET Q4$
1180 IF Q4$="1" THEN 1210
1190 IF Q4$="2" THEN 1210
1200 GOTO 1170
1210 HOME
1220 VTAB 4:HTAB 16:PRINT
"*MENU.2*"
1230 VTAB 5:HTAB 16:PRINT
"====="
1240 VTAB 10:HTAB 10:PRINT
"1.LISTA DE VALORES DE S"
1250 VTAB 13:HTAB 10:PRINT
"2.REPRESENTACAO DE S EM HGR"
1260 VTAB 16:HTAB 10:PRINT
"3.LISTA E HGR"
1270 VTAB 22:INVERSE:PRINT "A
SUA ESCOLHA":NORMAL
1280 VTAB 22:HTAB 25:GET Q5$
1290 IF Q5$="1" THEN PRINT:
HOME:GOTO 1340

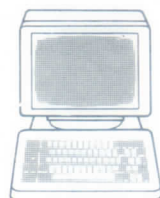
```



APPLE II



cont. PARA ESTAR "À CORRENTE"



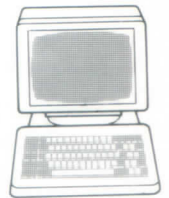
APPLE II

1300 IF Q5\$="2" THEN PRINT: HOME:GOTO 1340	1730 HPLOT X,Y	
1310 IF Q5\$="3" THEN PRINT: HOME:GOTO 1340	1740 NEXT RD	
1320 GOTO 1280	1750 FOR RD=1 TO 12	
1330 REM LISTA DOS S	1760 READ X	
1340 IF Q4\$="1" THEN PRINT ZZ\$;"PR#1"	1770 READ Y	
1350 IF Q5\$="2" THEN 1450	1780 Y=180+Y	
1360 IF Q4\$="1" THEN GOSUB 20000	1790 X=2+X	
1370 FOR F=FI TO FX STEP SF	1800 HPLOT X,Y	
1380 E=E+1	1810 NEXT RD	
1390 S=S(E)	1820 FOR RD=1 TO 14	
1400 PRINT "F=";F;" S";E;"="; S	1830 READ X	
1410 IF A=E THEN 1440	1840 READ Y	
1420 IF E/20=INT(E/20) THEN HTAB 29:VTAB 23:INVERSE:PRINT <<QUALQUER TECLA>>":NORMAL: GET Q6\$	1850 X=260+X	
1430 NEXT F	1860 Y=180+Y	
1440 IF Q4\$="1" AND Q5\$="1" THEN PRINT ZZ\$;"PR#0"	1870 HPLOT X,Y	
1450 IF Q5\$="1" THEN 2570	1880 NEXT RD	
1460 HOME	1890 FOR RD=1 TO 64	
1470 REM RESOLUCAO EM HGR2	1900 READ Q0	
1480 IF Q4\$="1" THEN POKE 1913,2:PRINT ZZ\$;"PR#1"	1910 NEXT RD	
1490 IF Q4\$="1" THEN GOSUB 20000	1920 FOR RD=1 TO 20	
1500 HGR2:HCOLOR=3	1925 READ X	
1510 A=A-1	1930 READ Y	
1520 DR=0	1935 X=2+X	
1530 PS=0	1940 Y=10+Y	
1540 ON Q3 GOSUB 30000,25000	1950 HPLOT X,Y	
1550 ON Q3 GOSUB 35000,40000	1960 NEXT RD	
1560 HS=170-(INT(S(PS+1))* 100)	1970 FOR RD=1 TO 109	
1570 FOR LI=1 TO 190 STEP 3	1980 READ X	
1580 HPLOT JJ,LI	1990 READ Y	
1590 NEXT LI	2000 X=JJ+X	
1600 IF Q3=2 THEN 2060	2010 Y=3+Y	
1610 FOR RD=1 TO 17	2020 HPLOT X,Y	
1620 READ X	2030 NEXT RD	
1630 READ Y	2040 RESTORE	
1640 X=JJ+X	2050 GOTO 2530	
1650 Y=80+Y	2060 FOR RD=1 TO 17	
1660 HPLOT X,Y	2070 READ X	
1670 NEXT RD	2080 READ Y	
1680 FOR RD=1 TO 41	2090 X=JJ+X	
1690 READ X	2100 Y=50+Y	
1700 READ Y	2110 HPLOT X,Y	
1710 X=2+X	2120 NEXT RD	
1720 Y=165+Y	2130 FOR RD=1 TO 41	
	2140 READ X	
	2150 READ Y	
	2160 X=2+X	
	2170 Y=61+Y	
	2180 HPLOT X,Y	
	2190 NEXT RD	
	2200 FOR RD=1 TO 12	
	2210 READ X	
	2220 READ Y	
	2230 X=2+X	
	2240 Y=3+Y	
	2250 HPLOT X,Y	



cont. PARA ESTAR "À CORRENTE"

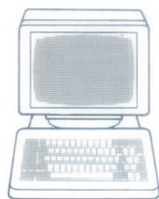
02260 NEXT RD	10100 X=LOG(S)
2270 FOR RD=1 TO 14	10200 Z=LOG(10)
02280 READ X	10300 Y=(1/Z)*X
2290 READ Y	10400 S=10*Y
2300 X=265+X	10500 RETURN
02310 Y=3+Y	15000 S=(1+(2*Q*D)^2)^.5
2320 HPLOT X,Y	15100 RETURN
02330 NEXT RD	20000 PRINT A\$
2340 FOR RD=1 TO 32	20100 PRINT AA\$
02350 READ X	20200 PRINT:PRINT
2360 READ Y	20300 PRINT "RESISTENCIA DE "
2370 X=2+X	;R;"OHMS"
02380 Y=165+Y	20400 PRINT
2390 HPLOT X,Y	20500 PRINT "INDUTANCIA DE "
02400 NEXT RD	;LL;"MICROHENRI"
2410 FOR RD=1 TO 40	20600 PRINT
02420 READ Q0	20700 PRINT "CAPACIDADE DE "
2430 NEXT RD	;CC;"PICOFARAD"
2440 FOR RD=1 TO 109	20800 PRINT
02450 READ X	20900 PRINT "FREQUENCIA DE
2460 READ Y	RESSONANCIA DE ";F0\$;"Khz"
02470 X=JJ+X	21000 PRINT
2480 Y=150+Y	21100 PRINT "BANDA PASSANTE A
02490 HPLOT X,Y	-3Db ";B\$;"Khz"
02500 NEXT RD	21200 PRINT
2510 RESTORE	21300 PRINT "FACTOR DE
02520 REM	QUALIDADE=";Q0
2530 PRINT CHR\$(17)	21400 PRINT
02540 IF Q4\$="1" THEN PRINT	21500 IF Q3=2 THEN PRINT
ZZ#"PR#0":POKE 1913,255	"VALORES DE 'S' EM Db=";GOTO
2550 GET Q7\$	21700
02560 TEXT:HOME	21600 PRINT "VALORES DE 'S'
2570 REM MENU DE FIM	DA RELACAO I/I0:"
02580 HOME	21700 PRINT
2590 HTAB 13:PRINT "*MENU DE	21800 PRINT "FREQUENCIA
FIM*"	MINIMA (FI) DE=";FI/1000;
2600 HTAB 13:PRINT "*****	"Khz"
*****"	21900 PRINT
02610 HTAB 10:VTAB 8:PRINT "1.	22000 PRINT "FREQUENCIA
RETORNO AO INICIO"	MAXIMA (FX) DE:"FX/1000;
02620 HTAB 10:VTAB 12:PRINT	"Khz"
"2. REIMPRESSAO DOS VALORES"	22100 PRINT
02630 HTAB 10:VTAB 16:PRINT	22200 PRINT "SALTOS DE
"3. FIM"	FREQUENCIA DE (SF) DE=";SF
2640 HTAB 5:VTAB 20:INVERSE:	22300 PRINT:PRINT
0PRINT "A SUA ESCOLHA(1,2 OU	22400 HOME
3)":NORMAL	22500 RETURN
02650 HTAB 30:VTAB 20:GET Q8\$	25000 FOR PP=1 TO 270 STEP
2660 IF Q8\$="1" THEN RUN	(270/A)
02670 IF Q8\$="2" THEN E=0:A=0:	25050 PS=PS+1
GOTO 860	25100 HT=INT(S(PS)*5)
2680 IF Q8\$="3" THEN HOME:END	25200 HS=INT(S(PS+1)*5)
02690 GOTO 2650	25300 HPLLOT PP,HT TO PP+(270/
2700 END	A),HS
010000 S=1+((2*Q*D)^2)	25350 HPLLOT PP,1



APPLE II



cont. PARA ESTAR "À CORRENTE"



APPLE II

○25400 IF S(PS+1)>S(PS) AND DR=0 THEN SP=S(PS-1):DR=34: JJ=PP	22, 4, 3, 5, 4, 5, 5, 5, 9, 5, 10, 5, 11, 5, 14, 5, 15, 5, 20, 5, 21, 5
○25500 NEXT PP	50007 DATA 22, 5, 23, 5, 24, 5, 3,
○26700 FOR LI=1 TO 270 STEP 2	6, 6, 6, 16, 6, 19, 6, 22, 6, 25, 6, 3,
○26800 HPLOT LI, 71	7, 4, 7, 5, 7, 13, 7, 14, 7, 15, 7, 20,
○26900 NEXT LI	7, 21, 7, 22, 7, 23, 7, 24, 7
○29000 RETURN	50010 REM DATA FI (12)
○30000 FOR PP=1 TO 270 STEP (270/A)	50011 DATA 3, 7, 7, 7, 3, 6, 7, 6, 3,
○30050 PS=PS+1	5, 4, 5, 5, 5, 7, 5, 3, 4, 4, 3, 5, 3, 6, 3
○30100 HT=190-(INT(S(PS)*5))	50015 REM DATA FX (14)
○30150 HS=190-(INT(S(PS+1)*5))	50016 DATA 3, 7, 7, 7, 9, 7, 3, 6, 8,
○30200 HPLOT PP, HT TO PP+(270/ A), HS	6, 3, 5, 4, 5, 5, 5, 7, 5, 9, 5, 3, 4, 4,
○30250 IF S(PS+1)>S(PS) AND DR=0 THEN SP=S(PS):DR=34: JJ=PP	3, 5, 3, 6, 3
○30275 HPLOT PP, 189	50020 REM DATA I/I0 (32)
○30300 NEXT	50021 DATA 5, 5, 6, 5, 7, 5, 6, 6, 6,
○30800 FOR LI=1 TO 270 STEP 2	7, 6, 8, 5, 9, 6, 9, 7, 9, 4, 11, 5, 11,
○30900 HPLOT LI, 175	6, 11, 7, 11, 8, 11, 5, 13, 6, 13, 7,
○31000 NEXT LI	13, 6, 13, 6, 14, 6, 15
○31010 FOR LI=190 TO 10 STEP -5	50022 DATA 9, 15, 10, 15, 11, 15,
○31020 HPLOT 0, LI	6, 16, 9, 16, 11, 16, 5, 17, 6, 17, 7,
○31030 HPLOT 2, LI	17, 9, 17, 10, 17, 11, 17
○31040 NEXT LI	50025 REM DATA Db (20)
○31050 FOR LI=190 TO 10 STEP -50	50026 DATA 6, 3, 8, 3, 6, 4, 8, 4, 4,
○31060 HPLOT 3, LI	5, 5, 5, 6, 5, 8, 5, 9, 5, 10, 5, 3, 6, 6,
○31070 NEXT LI	6, 8, 6, 11, 6, 4, 7, 5, 7, 6, 7, 8, 7, 9,
○31900 SP=190-(5*(INT(SP)))	7, 10, 7
○32000 FOR LI=15 TO 255 STEP 3	50030 REM DATA CURVA S (109)
○32200 HPLOT LJ, SP	50031 DATA 4, 3, 5, 3, 8, 3, 9, 3,
○32300 NEXT	12, 3, 15, 3, 17, 3, 18, 3, 19, 3, 22,
○32400 RETURN	3, 23, 3, 24, 3, 27, 3, 28, 3, 29, 3,
○35000 HPLOT 1, 0 TO 1, 190 TO 276, 190	32, 3, 33, 3, 34, 3, 37, 3, 38, 3, 39,
○35100 HPLOT 1, 0 TO 4, 4:HPLLOT 272, 186 TO 276, 190	3, 43, 3, 44, 3, 45, 3
○35600 RETURN	50032 DATA 3, 4, 7, 4, 10, 4, 12, 4,
○40000 HPLOT 1, 0 TO 276, 0	15, 4, 17, 4, 20, 4, 22, 4, 25, 4, 27,
○40100 HPLOT 1, 0 TO 1, 190	4, 32, 4, 35, 4, 37, 4, 43, 4, 3, 5, 7,
○40300 HPLOT 1, 190 TO 4, 186	5, 10, 5, 12, 5, 15, 5, 17, 5, 20, 5,
○40400 HPLOT 276, 0 TO 272, 4	22, 5, 23, 5, 24, 5, 27, 5, 28, 5, 32,
○40500 RETURN	5, 35, 5, 37, 5, 38, 5, 43, 5, 44, 5,
○50000 REM DATA F0 (17)	45, 5
○50001 DATA 4, 3, 5, 3, 6, 3, 3, 4, 3, 5, 4, 5, 5, 5, 7, 5, 8, 5, 9, 5, 3, 6, 7, 6, 9, 6, 3, 7, 7, 7, 8, 7, 9, 7	50033 DATA 3, 6, 7, 6, 10, 6, 12, 6, 15, 6, 17, 6, 18, 6, 19, 6, 22, 6, 25, 6, 27, 6, 32, 6, 35, 6, 37, 6, 45, 6, 3, 7, 7, 7, 10, 7, 12, 7, 15, 7, 17, 7, 19, 7, 22, 7, 25, 7, 27, 7, 32, 7, 35, 7, 37, 7, 45, 7
○50005 REM DATA B-3Db (41)	50034 DATA 4, 8, 5, 8, 8, 8, 9, 8, 13, 8, 14, 8, 17, 8, 20, 8, 22, 8, 23, 8, 24, 8, 27, 8, 28, 8, 29, 8, 32, 8, 33, 8, 34, 8, 37, 8, 38, 8, 39, 8, 43, 8, 44, 8, 45, 8
○50006 DATA 3, 3, 4, 3, 5, 3, 13, 3, 14, 3, 15, 3, 22, 3, 3, 4, 6, 4, 16, 4,	

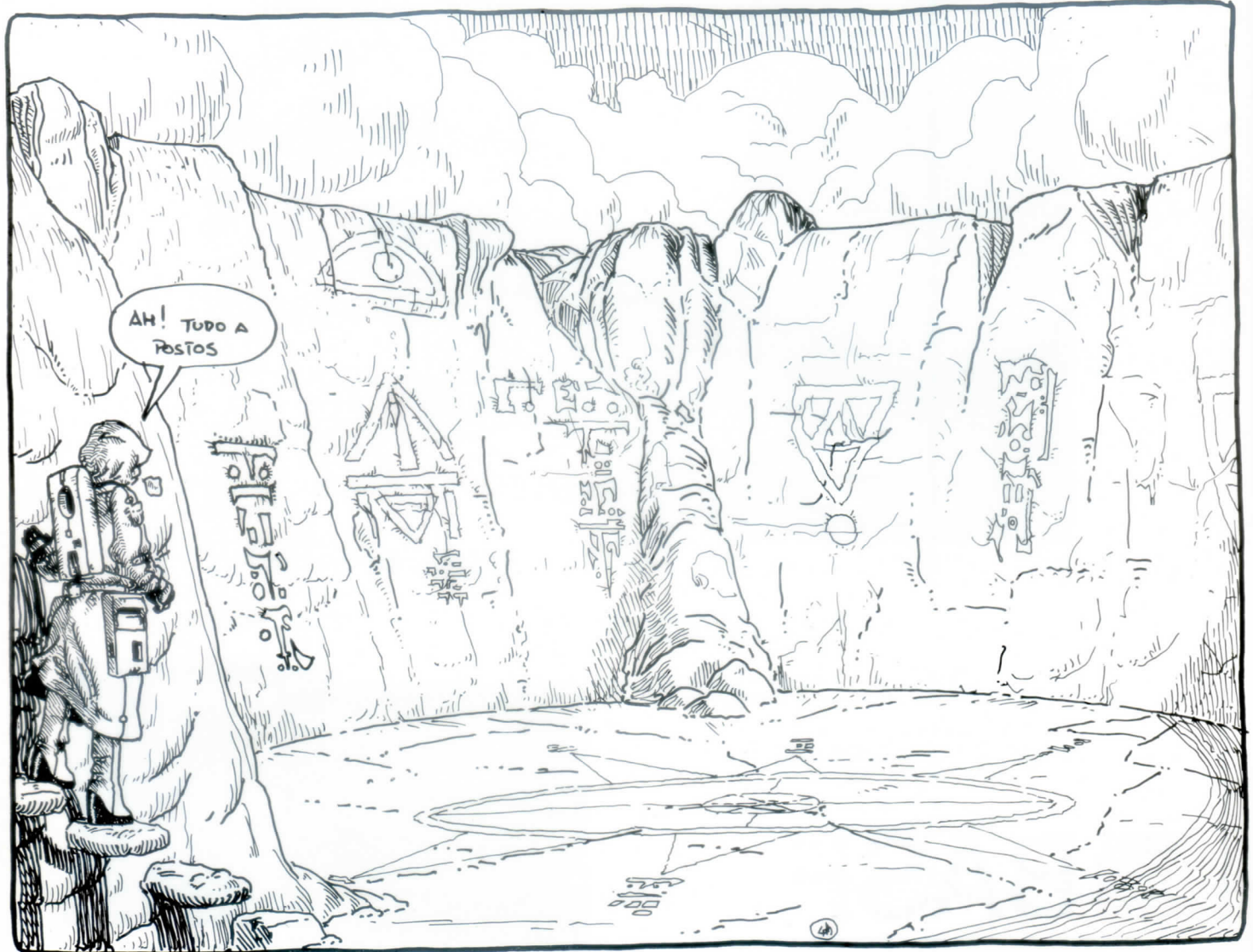
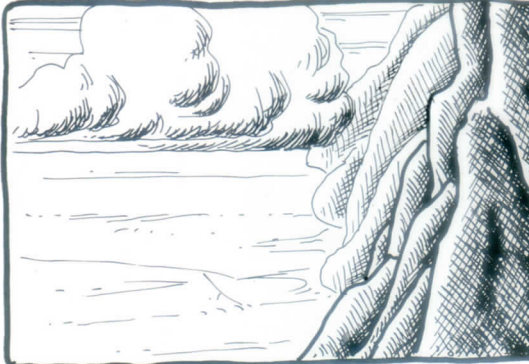
Go to **BASF FlexyDisk®**

***Tecnologia de Ponta
para a Sua Segurança.***



BASF







Bon... Proceeder por ordem...

AGORA A FRASE...

MY POWER IS RICH SK...

MAGIA



cont. LOGO

que permitam a resposta a perguntas mais ou menos complicadas acerca dos dados introduzidos.

Cria-se assim uma verdadeira linguagem informática em que os elementos da linguagem deixam de ser, as palavras ou procedimentos do LOGO para passarem a ser os objectos, as relações entre eles e as consultas a este sistema de entidades. O "micro-mundo" deixou de ser a tartaruga, para passar a ser um sistema de entidades organizado em torno de re-

lações semânticas.

Em Portugal...

À semelhança do que tem vindo a ser feito em muitos países, também em Portugal o LOGO está a ser introduzido no ensino. Assim, no âmbito do **projecto Minerva** do Gabinete de Estudos e Projectos do Ministério da Educação, o LOGO foi introduzido no início do corrente ano lectivo em duas escolas primárias e duas escolas preparatórias. □

Bibliografia:

Papert, Seymour, "Mindstorms: Children, Computers, and Powerful Ideas"

Rich, Elaine, "Artificial Intelligence"

Watt, Daniel, "Learning with LOGO"

Logo Computer Systems & Les Systemes D'Ordinateur Logo Internationale, "Sinclair LOGO 1/2"

DESINEL SOFT

Gestão Integrada : - Contabilidade
- Facturação
- Stocks
- Contas Correntes
- Letras
- Tesouraria
- Vencimentos - Normal
- Construção Civil
- Indust. de Pesca

Aplicações específicas por encomenda

Versões também para Apple //

* APPLE é uma marca registada de Apple Computer, Inc.
* CP/M é uma marca registada de Digital Research, Inc.

CIRTECH

Z80 Card - O MUNDO CP/M NO SEU APPLE !!
(Wordstar, Multiplan, DBASE, MBASIC,...)

CP/M Plus - Idem para //c

Interfaces para impressoras c/ buffer até 64Kb.

80 colunas + 64Kb p/ //e.

Programadores de EPROM.

DESINEL

BARREIROS & FIGUEIREDO, Lda

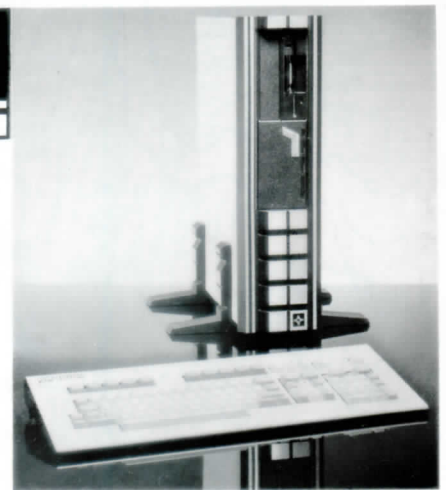
Av. Combatentes da Grande Guerra, 25 1º

3530 MANGUALDE

Tel. (032) 63591

U-MAN

32 BIT SUPERMICRO



3 PROCESSADORES - 68000 (10MHz)
- 6809 (I/O)
8035 (Keyboard)

AMPLA MEMÓRIA - RAM até 14Mb
- ROM até 512Kb

FLOPPY DISC - de 1 a 4 x 800Kb

HARD DISC - de 10 a 84Mb

DISPLAY - Mono ou Côr (RGB)
- Texto - 85 x 27
- Gráfico - 680 x 288

I/O - 2 RS232, 1 Centronics
- 10 bit A/D
- 4 slots de expansão
- Relógio c/ calendário e sintetizador de voz
- 16 portos programáveis I/O

MULTI-USER / MULTI-TASK

Alguns preços :
128Kb+2x800Kb - 564.500\$00
512Kb+800Kb+10Mb - 958.000\$00

Aceitamos propostas para revenda.



cont. BASIC

Uma outra vantagem, é que passa a ser possível a distinção das tarefas:

```
100 DIM V(5)
110 REM ENTRADA DE DADOS
120 FOR N= 1 TO 5
130 INPUT V(N)
140 NEXT N
150 REM PROCESSAMENTO
160 S=0
170 R=0
180 D=0
190 FOR N= 1 TO 5
200 S=S+V(N)
210 IF V(N) > 0 THEN R=R+V(N) ELSE D=D+V(N)
220 NEXT N
230 REM RESULTADOS
240 PRINT "BALANÇO", S
250 PRINT "RECEITAS", R
260 PRINT "DESPESAS", D
```

PASCAL

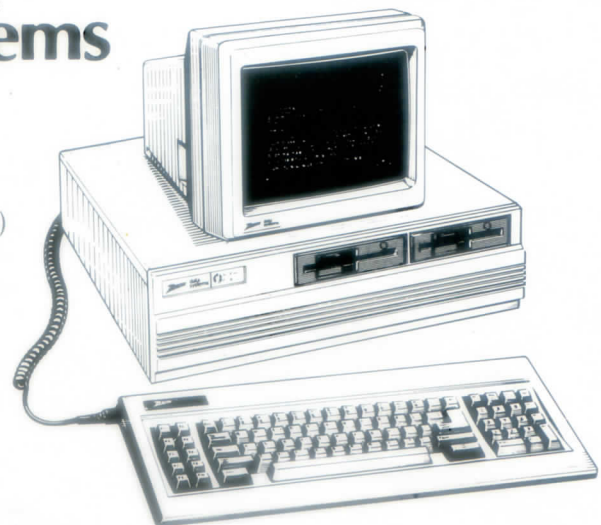
No último número, por lapso faltaram umas linhas no fim do segundo parágrafo. O seu conteúdo era: "Em Pascal a sequência é conseguida através de uma lista de instruções encaixadas entre um BEGIN, no início, e um END, no fim".

ZENITH

data systems

Z200

- Compatível IBM-AT/XT (Software/Hardware)
- Multiuser/Computador pessoal
- Rede local de computadores
- Alta resolução gráfica
- CPU-80286
- Sistemas operativos
 - Xenix 286
 - MS-DOS 3.1
- Memória
 - RAM desde 512K até 1.5MB
 - 1 a 3 unid. winchester de 20MB
 - 1 a 2 unid. floppy disk de 360K ou 1.2MB



Distribuidor para o Sul e Ilhas:

ABREU JUNIOR & CA. LDA*

RUA RODRIGUES SAMPAIO, 112-A - 1100 LISBOA - TELEF. 55 64 84

Distribuidor para o Norte

PD Prodata*

Centro de Processamento de Dados, s.a.r.l.

RUA AZEVEDO COUTINHO, 39-7.º - 4100 PORTO - TELEFS. 69 61 82/69 59 29

COMO LIGAR 2 PROGRAMAS

P. COUPATEZ

O autor propõe-nos um artigo muito sucinto que permite ligar dois programas entre si a partir duma cassete, isto para o Dragon 32.

Este procedimento está publicado num livro inglês escrito por Ian SINCLAIR (1983) nas edições Granada: "The DRAGON 32 and how to make the most of it".

No entanto essas instruções estão incorrectas nos endereços dos ponteiros.

O sr. Coupatez corrigiu-as e quer partilhar esse seu trabalho com todos os utilizadores do Dragon 32.

Procedimento:

1/ 'RENUM' do 2.º Programa para um endereço superior ao do 1.º programa.

2/ 'CLOAD' do 1.º programa. ATENÇÃO: não teclar 'LIST' ou 'RUN'!!

3/ 'PRINT PEEK (25); PEEK (26) ENTER' e chama os valores afixados respectivamente: a e b . Normalmente $a = 30$ e $b = 1$.

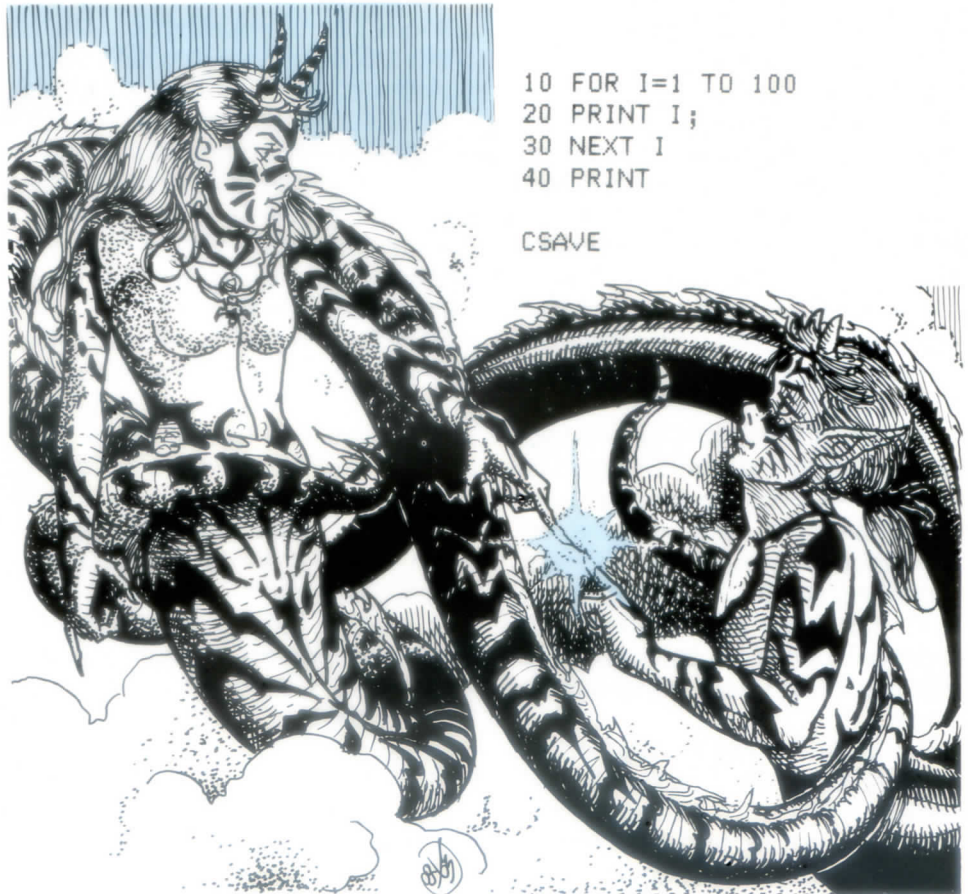
4/ TECLÉ:

```
'J=256*PEEK(27)+PEEK(28)-2  
ENTER'  
'H=INT(J/256):L=J-256*H  
ENTER'  
'POKE 25,H:POKE 26,J ENTER'
```

5/ 'CLOAD' o 2.º programa

6/ 'POKE 25, A: POKE 26, B ENTER' (sendo A e B os valores do ponto 3/).

Agora digite 'LIST' e vai ver que os seus 2 programas estão ligados!



1.º PROGRAMA.

```
10 FOR I=1 TO 100  
20 PRINT I;  
30 NEXT I  
40 PRINT
```

CSAVE

cf. modo operativo

Exemplo: 2.º PROGRAMA.

```
10 FOR I=100 TO 1 STEP -1  
20 PRINT I;  
30 NEXT I
```

RENUM 40

```
40 FOR I=100 TO 1 STEP -1  
50 PRINT I;  
60 NEXT I
```

CSAVE

RESULTADO:

```
10 FOR I=1 TO 100  
20 PRINT I;  
30 NEXT I  
35 PRINT  
40 FOR I=100 TO 1 STEP -1  
50 PRINT I;  
60 NEXT I
```



ESTRUTURA DE DADOS

PAULO AMARAL

Estruturas de dados porquê?

Depois da leve(!) introdução a este assunto no artigo anterior, urge alguma clarificação. A extensão do tema e a falta de informação corrente pode ter suscitado algumas dúvidas.

Ora nós criamos os computadores para nos auxiliarem na resolução de certos (muitos) problemas, o que se torna possível através da manipulação pura e simples de grandes quantidades de dados. Estes dados, embora abstractos, é que possibilitam o tratamento computadorizado dessa mesma informação.

Primeiro que tudo há que **analisar** o problema e decidir quais os objectos que o representam, seleccionando os essenciais e esquecendo os pouco importantes. A representação dos problemas no computador (através dos dados) significa, antes de mais, uma simplificação desses mesmos problemas.

Depois temos de decidir como representar a informação. É uma decisão difícil(!), sobretudo se aquela for muito volumosa e estiver inter-relacionada de diversas formas.

1.º passo — escolher uma linguagem

2.º passo — escolher e definir os tipos de dados

3.º passo — estruturá-los (definir estruturas)

4.º passo — programar

Se escolhermos a linguagem PASCAL, temos à nossa disposição:

- CONSTANTE
- VARIÁVEL
- EXPRESSÃO
- FUNÇÃO

para poder manipular os tipos fundamentais de dados.

Estes tipos ("TYPE") "caracterizam essencialmente o conjunto de valores a que uma constante pertence, ou que podem ser assumidos por uma variável ou expressão, ou ser gerados por uma fun-

ção" (*).

Depois é necessária a utilização de métodos de estruturação que tornem possível e aumentem a eficiência da computação. Muito antes do aparecimento dos computadores, por volta de 1750, já Euler dizia: "... se mais uma vez arregaçar as mangas para computar durante três dias e três noites seguidas, gastarei antes disso um quarto de hora para pensar nos métodos (de computação) que sejam mais apropriados."

No artigo anterior debrucei-me sobre o método de estruturação (não confundir com estrutura) mais frequentemente utilizado: o ARRAY. Prometi também introduzir os ponteiros (POINTERS). Vamos então enquadrar estes métodos segundo o PASCAL e o seu precursor: NIKLAUS WIRTH.

O PASCAL possui quatro métodos estáticos de estruturação:

- ARRAY
- RECORD
- SET
- FILE

Permite também a criação de estruturas dinâmicas como as listas ligadas, os anéis, as árvores e os grafos, através dos ponteiros (previstos na linguagem). É claro que não posso falar de tudo isto, mas para os mais interessados aqui fica a referência. Um conselho: dêem uma vista de olhos pela bibliografia!

Com esta pequena introdução tenciono mostrar como é simples e bela esta ciência. Bom. Deixemo-nos de filosofias e passemos à acção.

PROBLEMINHA

Temos uma lista de nomes ordenada alfabeticamente e colocada num array A

A: Índice	valor
1	CARLOS
2	FERNANDO
3	LUIS
4	PAULO
5	RAUL

Quer inserir-se mais um nome na lista: DIONISIO.

Procedimento:

1 — Procurar o sítio onde encaixa DIONISIO. É entre CARLOS e FERNANDO, isto é, entre os índices 1 e 2 (de preferência com um algoritmo eficiente como a busca binária).

2 — Abrir um espaço para encaixar o nome

A: Índice	valor
1	CARLOS
2	
3	FERNANDO
4	LUIS
5	PAULO
6	RAUL

Tivemos de chegar os quatro nomes (índice 2 a 5) uma casa para a frente (3 a 6).

3 — Inserir finalmente o nosso elemento

A: Índice	valor
1	CARLOS
2	DIONISIO
3	FERNANDO
4	LUIS
5	PAULO
6	RAUL

Duma forma semelhante para apagar este nome da lista temos de chegar para trás todos os elementos do índice 3 ao 6 para 2 a 5.

A limitação desta estrutura está em que tivemos de mover 4 elementos para a frente e para trás. Se isto não parecer uma limitação, então pensemos em termos reais numa lista com 1000 elementos e com 30 caracteres cada um, por exemplo. A quantidade média de memória a deslocar neste caso seria o correspondente a $500 \times 30 = 15000$ caracteres...

A estrutura que vamos estudar hoje melhora substancialmente o procedimento, tornando-o independente do volume dos elementos. Para isso vou introduzir primeiro o conceito de ponteiro.

cont. ESTRUTURA DE DADOS

POINTER

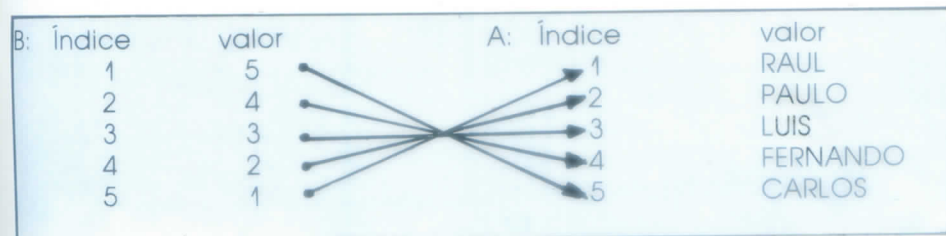
Esta palavra é um estrangeirismo. A sua tradução é ponteiro. Entre nós, informáticos, chamamos-lhe muitas vezes POINTER, bem à inglesa. Passe, portanto, o estrangeirismo.

Tradicionalmente o ponteiro é um índice (um número) que traduz a localização de um determinado elemento (pode ser até outro ponteiro). Ou seja, ao utilizarmos um ponteiro podemos aceder indirectamente aos nossos dados. Convém, desde já, deixar claro que em PASCAL o conceito de ponteiro é ainda mais abstracto e geral, relacionando-se directamente com estruturas dinâmicas. Para não complicar demasiado as coisas, não vou esgotar o assunto e vamos implementar os ponteiros em ARRAY's. É que ainda estamos a lidar com estruturas estáticas. No próximo artigo vou dar o primeiro cheirinho de estruturas dinâmicas, e aí, clarificarei melhor o problema. Adiante.

No BASIC, que é uma linguagem menos evoluída, a única maneira de implementar estruturas com ponteiros é justamente com ARRAY's, e é, em tudo, semelhante às linguagens de baixo nível (linguagem máquina e assembly).

Resumindo: Vamos utilizar um array cujos elementos são os índices de outro array que tem os elementos principais (nomes).

Antevisão com uma ordem alfabética inversa:



Como já repararam o grafismo do ponteiro é



Vamos agora implementar a estrutura de dados do exemplo an-

terior com este sistema.

** LISTA ORDENADA INDEXADA **

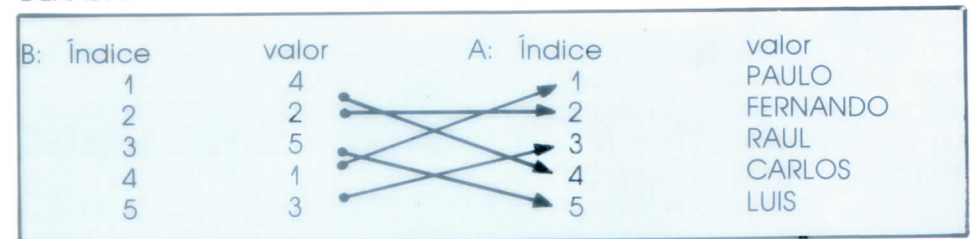
O array que tem os ponteiros não é mais do que uma tabela de índices. Este processo é utilizado em muitos sistemas e, por vezes, chamam-lhe directórios (os célebres). Vamos adaptar o exemplo anterior segundo este processo.

Primeiro, é criada uma lista de nomes ordenada dum forma totalmente arbitrária, por exemplo de acordo com a chegada cronológica dos dados (situação muito comum no dia a dia; é raro chegarem por ordem alfabética).

Ex:

A: Índice	valor
1	PAULO
2	FERNANDO
3	RAUL
4	CARLOS
5	LUIS

À medida que o conjunto dos nomes é criado (simples lista desordenada) organizamos a tabela de índices que mantém a ordem alfabética. É assim que concretizamos uma LISTA ORDENADA E INDEXADA.



Vamos partir daqui e proceder às operações fundamentais:

- inserir um elemento
- apagar um elemento
- elemento será mais uma vez

DIONISIO.

1.º passo — utilizar uma busca para encontrar o ponto de inserção na lista. Para mostrar como se utiliza esta estrutura vou exemplificar essa busca:

A) O meio da lista é $(5 + 1)/2 = 3$. O 3.º elemento do array B é 5. O 5.º elemento do array A é LUIS. Ora alfabeticamente DIONISIO < LUIS. Escolhe-se a metade inferior.

B) Escolhe-se $(3 + 1)/2 = 2$. Da mesma maneira o 2.º elemento do array B é 2.

O 2.º elemento do array A é FERNANDO. DIONISIO < FERNANDO: escolhe-se a metade inferior.

C) Escolhe-se $INT((2 + 1)/2) = 1$. O 4.º elemento do array A é CARLOS. Como DIONISIO > CARLOS o índice de DIONISIO vai ter que ser encaixado entre o primeiro e segundo índice do array de ponteiros B.

Até agora o trabalho foi praticamente o mesmo requerido pela estrutura anterior, com a diferença de que a localização dos nomes foi feita indirectamente. A vantagem vem a seguir.

2.º passo — Para a inserção, em vez de se movimentarem todos os nomes, movimentam-se apenas os apontadores, o que é suficiente para manter a ordem alfabética. Assim, acrescento sem preocupações o nome DIONISIO ao array A no espaço imediatamente disponível, enquanto inscrevo o índice 6 no lugar que lhe compete no array B.



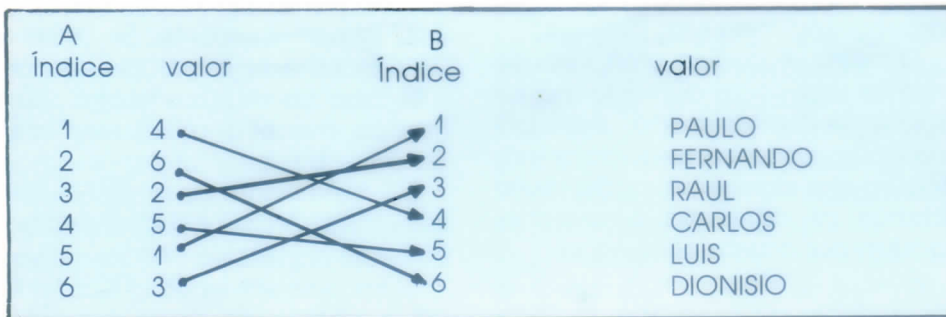
cont. ESTRUTURA DE DADOS

B: 4 2 5 1 3 valor
 1 2 3 4 5 índice

4 2 5 1 3 valor
1 2 3 4 5 6 índice

4 6 2 5 1 3 valor
1 2 3 4 5 6 índice

Vem finalmente



Para se apagar um elemento o processo é semelhante

1.º passo — (opcional) levar o dito elemento a 0

2.º passo — apagar o apontador

3.º passo — chegar todos os apontadores seguintes uma casa para trás

Nota: Só é preciso ter cuidado em memorizar os espaços que vão ficando vagos para não haver o risco de se ter "lixo" no array A.

Um bom método será alterar o 2.º passo: em vez de se apagar o apontador definitivamente, faz-se uma transferência para a parte superior e vaga do array B. Podem ficar assim memorizados todos os espaços que vão ficando vagos. Depois, quando se for atribuir um espaço para um novo elemento, verifica-se primeiro se há algum livre no interior do conjunto da lista dos nomes. Existem mais sistemas para isto, é claro. Para mim, o melhor deles é o das listas ligadas: uma estrutura dinâmica, das mais simples, que tenciono explicar no próximo artigo.

Resta-me fazer um pequeno estudo de eficiência para provar a melhoria conseguida por esta es-

trutura. Já vimos que seria necessário deslocar em média 15000 caracteres com a tal lista de 1000 nomes de 30 caracteres cada um, com a estrutura da lista ordenada (simples), sempre que quiséssemos proceder a uma operação fundamental. Com esta estrutura indexada basta deslocar apenas os ponteiros (1000/2 = 500 em média).

Aqui cada ponteiro pode ser codificado (internamente ou pelo programador) em dois Byte's, portanto deslocar-se-á apenas 1000 caracteres. E o processo é independente do tamanho dos elementos da lista principal...

Conclusão: se a quantidade de dados for pequena, esta estrutura não traz grandes vantagens, mas, se, pelo contrário, lidarmos com muitos dados, é aconselhável experimentar os grandes benefícios de velocidade de processamento que esta estrutura, ainda simples, já traz.

PEQUENO TEMA PARA REFLECTIR

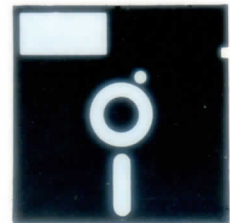
Até agora não foi feita qualquer referência ao tipo de memória que se pode utilizar: viva ou secundária (até porque se lidou com array's). Ora esta estrutura permite que já se possa ter um ficheiro (de acesso aleatório) de nomes, em memória secundária — só é necessária a presença em memória viva do array de ponteiros (B). Sugiro ao leitor que reflecta sobre a implementação de um tal sistema, que mais tarde abordarei. □

Bibliografia:

Algorithms + Data Structures = Programs, Niklaus Wirth

Introduction to the Design and Analysis of Algorithms, Goodman, Hedetniemi

Pascal for Programers, Olivier Lecarme, Jean Louis Nebut/Computer Data Structures, John Pfaltz



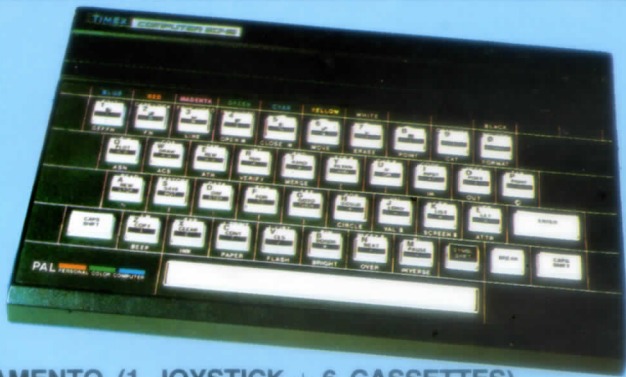
No NATAL...
ofereça
CONHECIMENTO
ofereça a
assinatura
da revista
SOFTWARE!

(*) NIKLAUS WIRTH

TC 2048/TC 2068

DOIS COMPUTADORES DISTINTOS

A MESMA TECNOLOGIA AO SERVIÇO DA MICRO-INFORMÁTICA



● CAMPANHA DE TROCAS

● OFERTA LANÇAMENTO (1 JOYSTICK + 6 CASSETTES)

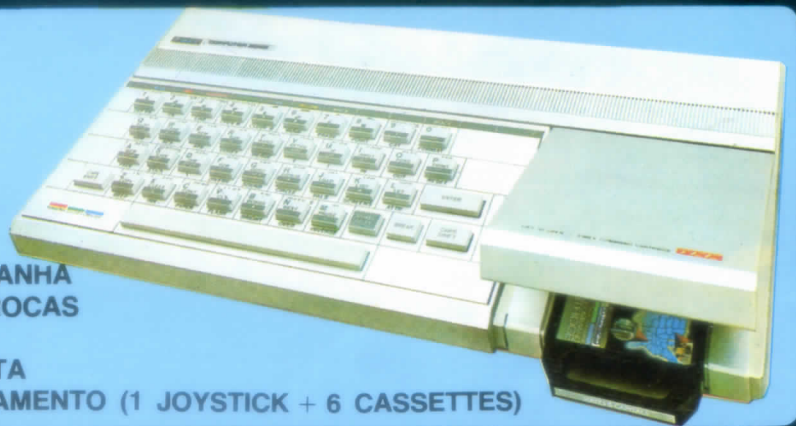
TIMEX COMPUTER 2048

- Compatível SPECTRUM
- Teclado circuito impresso
- Porto para Joystick tipo KEMPSTONE incorporado
- Saída para Monitor Vídeo composto
- Interruptor ON/OFF
- LED de sinalização
- Barra de espaços
- Ficha de expansão para periféricos com sinal R.G.B.
- Dupla resolução gráfica
- Sistema de disco — (pode suportar, controlando até 4 Drives de 3", 140 K/face (formatado), com porto RS 232 para comunicação/série)
- Armazenamento de Informação:
 - Gravador de cassettes
 - Microdrive
 - Sistema de Floppy Disk

TIMEX COMPUTER 2068

Na linha do TC 2048 e compatível com ele, apresenta-se mais potente graças à adição de novas funções, sintetizador de Som e Porto para Cartridges. Este permite-lhe carregar instantaneamente programas em ROM, sem necessidade de recorrer ao tradicional gravador de cassettes.

- Cartridges disponíveis:
 - ANDROIDS ● CRAZY BUGS ● BUDGETER
 - FLIGHT SIMULATOR ● CASINO
- brevemente
- PROCESSADOR DE TEXTO/GESTOR DE LEITOR DE CÓDIGO DE BARRAS
- Compatível SPECTRUM através de Cartridge emuladora
- Novas funções: STICK/ON ERR/FREE/DELETE/SOUND
- 3 Canais de som independentes e programáveis



● CAMPANHA DE TROCAS

● OFERTA LANÇAMENTO (1 JOYSTICK + 6 CASSETTES)

TABELA COMPARATIVA

	SPECTRUM +	TC 2048	TC 2068
Microprocessador	Z80-A	Z80-A	Z80-A
RAM física	48K	48K	48K
RAM utilizável	41472 Bytes	41472 Bytes	38652 Bytes Modo Spectrum 41772 Bytes
ROM	16K	16K	24K
Écran	24×32 ou 24×64	24×32, 24×64, 24×80	24×32, 24×64, 24×80
Resolução (Pixel)	256×192	256×192, 512×192	256×192, 512×192
Som	Beep	Beep	Beep e gerad. programável
Joysticks	Não	1 Joystick	2 Joystick
Solid State Software	Não	Não	Sim
Gravador	Vulgar	Vulgar	Vulgar
Disk Drive	Pode suportar	Pode suportar	Pode suportar
TV	UHF 36	UHF 36	UHF 36
Vídeo composto	Não	Sim	Sim
Interruptor ON/OFF	Reset	Sim	Sim
Cores	8 + Bright = 15	8 + Bright = 15	8 + Bright = 16
Teclado circuito impresso	Não	Sim	Sim
Circuitos integrados	26	15	15
Manual em português	Não	Sim	Sim
Garantia	6 meses	1 ano	1 ano

- Monitor vídeo composto monocromático (fósforo verde)
NEPTUN 156
- Impedância de entrada: 75 ohm
 - Resolução 520 Pontos/linha
 - Alimentação 220 VAC ou 12 VDC
 - Écran: 31 cm (12")

TIMEX PRINTER 2080 IMPRESSORA MATRICIAL

- Ligação a qualquer computador com saída RS232
- Papel A-4 ou banda contínua
- Cópias múltiplas
- Capacidade gráfica
- Velocidade 100 caract./seg.
- 7 categorias impressão gráfica
- 130 caracteres.

TIMEX

NCR DECISION MATE

